

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається

Завідувач кафедри

_____ Скарга-Бандурова І.С.

« _____ » _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження комп'ютерної системи керування рухомих об'єктом

Освітній ступінь “Магістр”

Спеціальність 123 “Комп’ютерна інженерія”

Науковий керівник роботи:

(підпис)

В.М.Барбарук

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О.Критська

(ініціали, прізвище)

Студент:

(підпис)

К.І.Матюшко

(ініціали, прізвище)

Група:

КІ-17зм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітній ступінь магістр

Напрямок підготовки _____

(шифр і назва)

Спеціальність 123 "Комп'ютерна інженерія"

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

І.С. Скарга-Бандурова

« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Матюшко Катерині Ігорівні

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження комп'ютерної системи керування рухомим
об'єктом

керівник проекту (роботи) Барбарук Віктор Миколайович, к.т.н., доц.

(прізвище, м. 'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «18» 10 2018 р. № 221/48

2. Строк подання студентом роботи 10.01.2018

3. Вихідні дані до роботи Матеріали науково-дослідної практики,
програмно-апаратні засоби систем відеоспостереження

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити) Огляд технологій комп'ютерного зору, методи оцінки датчика

комп'ютерного зору kinect, результати тестування Kinect, інтерфейс kinect-

робот та результати тесту, охорона праці та безпека в надзвичайних ситуаціях,
висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 18.10.2018

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Розробка технічного завдання	10.09.2018-15.09.2018	
2	Аналіз завдання, робота з літературою	16.09.2018-22.09.2018	
3	Аналіз датчику Kinect	26.09.2018-06.10.2018	
4	Виконання та аналіз результатів тестування датчику Kinect	07.10.2018-25.11.2018	
5	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2018-1.12.2018	
6	Оформлення пояснювальної записки, автореферату та презентації	2.12.2018-09.01.2019	
7			

Студент

_____ (підпис)

Матюшко К.І.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Барбарук В.М.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Матюшко К.І. Дослідження комп'ютерної системи керування рухомим об'єктом.

Метою атестаційної роботи є узагальнити теорію управління і принципи побудови систем автоматичного управління для будь-яких рухомих об'єктів. У ході виконання атестаційної роботи проаналізовані можливості та роль датчика Kinect в комп'ютерній системі автоматичного керування об'єктом, що рухається.

Ключові слова: kinect, комп'ютерний зір, робот, карта глибини, точність, рухомий об'єкт, ігри, тести.

ABSTRACT

Matyushko K.I. Research of a computer system of control of a moving object.

The major goal of this thesis is to generalize the theory of management and principles of the construction of automatic control systems for any movable objects.

In order to analyze the capabilities and role of the Kinect sensor in the system of computer automatic control of the moving object.

Keywords: kinect, computer vision, robots, depth maps, accuracy, moving objects, games, tests.

АННОТАЦИЯ

Матюшко К.И. Исследование компьютерной системы управления подвижным объектом.

Целью аттестационной работы является обобщить теорию управления и принципы построения систем автоматического управления для любых подвижных объектов. В ходе выполнения аттестационной работы проанализованы возможности и роль датчика Kinect в компьютерной системе автоматического управления движущимся объектом.

Ключевые слова: kinect, компьютерное зрение, робот, карта глубины, точность, движущийся объект, игры, тесты.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І	
ТЕРМІНІВ	6
ВСТУП	7
1 ОГЛЯД ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ	9
1.1 Ігрові технології.....	9
1.2 Робототехніка, як наука.....	12
1.3 Оцінка впливу ігрових технологій на робототехніку.....	15
1.4 Приклади використання робототехніки в медицині	16
1.5 Технологія 3D-зображень.....	17
1.6 Дослідження сенсору комп'ютерного зору Kinect	18
1.7 Порівняння з іншими датчиками	24
2 МЕТОДИ ОЦІНКИ ДАТЧИКА КОМП'ЮТЕРНОГО ЗОРУ KINECT	25
2.1 Методи тестування та налаштування.....	25
2.2 Методи тестування роздільної здатності.....	26
2.3 Методи тестування стабільності і точності	27
2.4 Accuracy	28
2.5 Модель реконструкції.....	31
2.7 Кут зору.....	34
2.8 Частотна характеристика Kinect та пропускна здатність	35
3 РЕЗУЛЬТАТИ ТЕСТУВАННЯ KINECT	37
3.1 Роздільна здатність	37
3.2 Стабільність	39
3.4 Точність глибини.....	43
3.5 Точність XY	44
3.6 Точність на краю	45
3.7 Реконструкція моделі	45
3.8 Кут огляду.....	48
3.9 Частотний відгук і пропускна здатність Kinect.....	49
3.10 Інші спостереження	52
4 ІНТЕРФЕЙС KINECT-РОБОТУ ТА РЕЗУЛЬТАТИ ТЕСТУ.....	53
4.1 Робот	53
4.2 Kinect - інтерфейс робота.....	54

4.3 Kinect - пропускна здатність робота системи.....	56
4.4 Демонстрація того, як Kinect безпосередньо контролює положення робота.....	57
4.5 Kinect - пропускна здатність системи роботів.....	58
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	62
5.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектowanego об'єкту, що мають вплив на персонал.....	62
5.2 Заходи щодо техніки безпеки.....	63
5.3 Заходи, що забезпечують виробничу санітарію і гігієну праці	66
5.4 Рекомендації по пожежній безпеці	69
5.5 Вплив на навколишнє середовища	71
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	75
ДОДАТОК А. KINECT-ROBOT ІНТЕРФЕЙС	78
ДОДАТОК Б. ЕЛЕКТРОННІ ПЛАКАТИ	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

САУ – система автоматичного керування

3D – 3-dimensional

MIT – Массачусетський технологічний інститут

HD – high definition

ІЧ – інфрачервоний

SDK – software development kit

MATLAB – Matrix Laboratory

ROM – Read-only Memory

OpenNI – Open Natural Interaction

PCL – Point Cloud

RGBD – Red Green Blue Depth

ВСТУП

У зв'язку зі збільшенням кількості транспортних засобів, інтенсивності руху, а також збільшенням рівня навколишніх наших технологій стає актуальним питання автоматизації управління рухомими об'єктами (зокрема автомобілями, військовою та спеціальною технікою в екстремальних умовах). Це дозволить вирішити відразу кілька проблем сучасності:

- кардинальна мінімізація аварій і практично повне виключення людських жертв, а значить і витрат на страхування і медицину;
- зниження вартості транспортування вантажів і людей за рахунок економії на заробітній платі і часу відпочинку водіїв, пілотів та обслуговуючого персоналу, а також економії палива;
- підвищення ефективності використання транспортних шляхів за рахунок централізованого управління потоком;
- поява можливості самостійно пересуватися на транспортному засобі без спеціального посвідчення;
- економія часу, що витрачається на управління об'єктом, що рухається;
- перевезення вантажів та особового складу під час природних і техногенних катастроф або військових дій.

Однак у автоматизації управління і зворотна сторона: неможливість перехоплення управління в критичних ситуаціях, або ж навпаки - відсутність досвіду у того, хто взяв управління, втрата приватності, оминання безпілотних транспортних засобів, постановка етичного питання про вибір дії, яка принесе найменше за кількісними або якісними (наприклад, на кону життя однієї дитини або двох дорослих) число жертв.

Один із найважливіших складових елементів системи автоматичного керування є датчики комп'ютерного зору та сенсори, які можуть відновити навколишнє 3D-середовище, для прийняття наступного рішення, щодо руху.

У статті "Використання Microsoft Kinect і обчислювальних можливостей потокових процесорів для проведення занять у віртуальному світі vAcademia" розглядається застосування сенсора Kinect для проведення занять в тривимірному освітньому віртуальному світі. Система дозволяє захоплювати жести, положення тіла і голови викладача і перетворювати їх в 3D-анімацію аватара лектора.

Одна з найважливіших особливостей датчика Kinect - відновлення 3D-сцени простору, що знаходиться в полі зору пристрою. Однією з альтернативних технологій в

даному напрямку є стереозрення. Воно засноване на роботі людського зору - на об'єкт спрямовані дві еталонних камери, і потім за допомогою проєктивної геометрії обчислюється відстань до об'єкта або відновлюється 3D-модель простору. Але у цієї технології є свої недоліки: неможливість коректно відновлювати простір, яке знаходиться за прозорими об'єктами, такими як вода, наприклад. Також існують лазерні далекоміри. Вони позбавлені недоліку стереозрення, але при цьому не можуть передати інформацію про яркостной характеристиці сцени, маю велике енергоспоживання і високу вартість.

Kinect так само отримує хмару точок, яке за допомогою відповідного програмного забезпечення можна перетворити хмару точок характерну для лазерних далекомірів, тобто є дешевою заміною. Сенсор має звичайну цифрову камеру, що дозволяє при накладенні фотографії на карту глибини простору відновити характеристику яскравості.

Як висновок можна сказати, що сенсор Kinect є унікальним пристроєм, який можна ефективно використовувати прототипом комп'ютерного зору в робототехніці, 3D-скануванні, розпізнаванні жестів, міміка й мови, в системах спостереження і безлічі різних сфер. Основна його перевага над іншими пристроями машинного зору - ціна і доступність як для професіоналів, так і для любителів-ентузіастів.

Мета даної магістерської атестаційної роботи – узагальнити теорію управління і принципи побудови систем автоматичного управління для будь-яких рухомих об'єктів на основі комп'ютерного зору.

Завдання – ґрунтуючись на теоретичній частині даної роботи, проаналізувати можливості та роль датчика Kinect в системі автоматичного керування об'єктом, що рухається.

1 ОГЛЯД ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

Датчик Kinect від Microsoft був призначений як ігровий пристрій для консолі Xbox 360. Але це зайняло багато часу, перш ніж робототехнічні дослідники виявили потенціал у Kinect. З Kinect, технологія найсучаснішої глибини відчуття стала раптово доступна за низькою ціною.

1.1 Ігрові технології

Перша відома електронна гра прийшла ще в 1947 році, коли Томас Т. Гольдсміт-молодший і Естле Рей Ман побудували те, що вони називають "катодно-променевою трубкою". [3] Гра була імітатором ракети, де метою гравця було збиття літаків на дисплеї, яке виглядало як екран радарів світової війни 2. Графіка була поганою, а літаки були вручну забарвлені на екран.

Чотири роки потому в 1951 році винахідник Ральф Баер висунув ідею використання телевізійного екрана як основи ігрових технологій. Але коли він передав ідею своєму керівництву в компанії, у якій він працював, ідея була відхилена.

Пізніше в 1952 р. доктора філософії в Кембриджському університеті О.Дуглас зробив те, що вважається першою цифровою графічною комп'ютерною грою. Він називався "ОХО" і був реалізацією гри "Tic-tac-toe". Гра може бути відтворена лише на одному комп'ютері в Кембриджі, тому популярність була досить обмежена.

У 1958 році гра "Теніс на двох" була розроблена W. Higinbotham. Гра була відтворена на дисплеї осцилографа та імітувала тенісний матч. Ця гра була першою, яка дозволила двом гравцям повністю контролювати рух на екрані. Гра запускається на аналоговому комп'ютері та розраховує гравітаційний ефект на м'яч, надавши йому правильні криві на шляху кульки. Ця гра була дуже популярною, і з'явилися довгі черги, які хотіли спробувати гру, коли працювала лабораторія Higinbotham на її виставці. На жаль, Higinbotham не бачив цінність у тому, що він створив і не прийняв патент на свій винахід.

У цей час більшість комп'ютерів знаходилися в університетах, і ігри, рідко відкривались на публіці, оскільки загалом лише їх творці грали в них, але в 1961 році група студентів MIT зробила гру називається "Spacewar!". Тут два гравці намагалися

стріляти один з одним двома космічними апаратами. "Spacewar!" Розповсюджувався через Інтернет і вважається першим широко доступним комп'ютерною грою. У 1969 році нова версія "Spacewar" дозволила двом гравцям грати один проти одного з різних комп'ютерів, це був перший випадок онлайн-ігор [4].

У 1966 р. Ральф Бар міг продовжувати свою роботу над своєю ідеєю відеоігор, тепер під таємним військовим проектом під назвою «Браун Бокс» [5]. Ідея полягала в тому, що гра може використовуватися під час підготовки військового персоналу. Цей план провалився, і в 1972 році концепція була продана Magnavox, яка називає її "Magnavox Odyssey" і розповсюджує її як першу в світі систему ігрових ігор. Електроніка була аналогом і машина була без будь-яких процесорів або пам'яті. Odyssey вивів електронні ігри з аркад в будинок. Це був початок того, що сьогодні стало багатомільйонною індустрією. Тільки в США індустрія відеоігор склала 12,5 млрд. дол. США у 2006 р. [44].

Ігрові консолі розміщуються в різних поколіннях, щоб розділити їх, залежно від того, з якої епохи часу вони знаходяться та які технології вони використовують. У наступному розділі буде використано ці покоління підрозділів, щоб описати, як розвивалася технологія відеоігор.

Перше покоління відеоігор почалося в 1972 році з виходом "Magnavox Odyssey". Ця консоль тримала ринок протягом декількох років, але не дуже успішно. Так було, аж поки Атарі не презентує домашню версію популярної аркадної гри "PONG", яку взяли домашні ігри. Консолі цього покоління могли в основному виконувати лише одну гру, оскільки ігри були "запрограмовані" безпосередньо на дискретній логіці консолей.

Odyssey мала можливість змінювати картриджі, які змінюють електричну схему і підтримувати різні ігри [2]. У цей час реальної графіки не було, і ігри склалися з квадратів, які рухалися навколо на екрані, щоб змінити фон ігор, на які гравці повинні були розміщувати пластикові накладки на своєму телевізійному екрані.

Друге покоління ігрових консолей вийшло в 1976 році з використанням чіпів і мікропроцесорів ROM (Read-only Memory). Ігри зберігалися на ПЗУ в картриджах, які можна було підключити до ігрових консолей, і мікропроцесор виконуватиме ігри на них [2]. Обсяг пам'яті був досить обмеженим: картриджі мали від 2 Кб пам'яті для перших консолей цього покоління та до 32 Кб для останніх консолей другого покоління. Ціни на RAM були високими і це вплинуло на консолі, що часто мали менше 1Кб оперативної пам'яті. Найчастіше процесори випускали 8-бітну графіку і працювали зі швидкістю від 1 до 4 МГц [6].

Ще одна річ, яку варто відзначити з цього періоду часу, - це введення першої гри з справжнім 3D-середовищем, це була аркадна гра "Battlezone" [44, с. xix].

Третє покоління ігрових консолей з'явилося в 1984 році, коли Nintendo вийшов на ринок консолі. Це покоління виробило кілька нових функцій порівняно з попереднім поколінням, за винятком більшої кількості пам'яті та обробки живлення в консолях, що дає кращу графіку та ігри. Найважливішою особливістю третього покоління, ймовірно, є введення геймпада як нового гра-контролера, який захопив старий джойстик та клавіатури. І, звичайно, гра Супер Маріо як один із найпопулярніших світових ігрових персонажів всіх часів [2].

Четверте покоління почалося приблизно в 1987 році, що привело нас до 16 бітних консолей, що дійсно покращило швидкість роботи консолей. Це покоління також представило нову технологію CD-ROM на ринку консольних ігор. У 1989 році "TurboGrafx-16" була першою консоллю, що постачалася з програвачем компакт-дисків. Перед компакт-диском ігри можуть складати максимум 256 КБ коду, але тоді CD міг вмістити до 550 МБ ігрового коду, що давало іграм значне покращення складності, деталізації та звуку [44, с. 119]

П'яте покоління почалося приблизно в 1993 році. У цьому поколінні консолі мали 32-розрядні процесори. Деякі навіть мали 64, як і популярний Nintendo 64. Більше потужності обробки і нова потужна технологія CD зробила 3D-ігри основним фокусом цього покоління.

Sony дебютувала на ринку консолі з їх першою консоллю, Playstation. Консоль Sony була першою консоллю, яка коли-небудь продавала більше 100 мільйонів одиниць [7].

Шосте покоління консолей почалося з "Dreamcast", що з'явилася в 1998 році і стала першою консоллю для вбудованого модему для онлайн-ігор. Як завжди від покоління до наступного, обчислення потужність збільшилася та графіка гри була вдосконалена. Окрім прийому онлайн-ігор на консолі, це покоління також використовувало популярні альтернативні контролери. Подібно гітарі у "Guitar Hero" та Playstation "EyeToy", один з перших успішних рухомих ігрових інтерфейсів [2]. Такі альтернативні контролери використовувались і в попередніх поколіннях, але технологія була досить неточною, і вони ніколи не стали дійсно популярними.

Сьоме покоління є нинішнім поколінням і почалося в 2005 році за допомогою Xbox 360, незабаром слідують Playstation 3 та Nintendo Wii. Коли Xbox і PlayStation зосередили увагу на якісній графіці HD і більшій обчислювальній потужності в консолі, ніж будь-коли раніше, Nintendo Wii застосував інший підхід. З метою залучення більш широкого кола геймерів у них було менше графіки та менших ігор, проте нові технології контролерів надавали їм величезну частину ігрового ринку. Контролер Wii включає 3D

детектор руху і дає гравцю абсолютно нову взаємодію з іграми, які вони грали.

Після успіху Wii, Playstation випустила свій контролер "Move" в 2010 році. Дві основні частини контролера "Move" - це камера "Playstation Eye" і ручна паличка. Система "Переміщення" може відстежувати рух тіла гравця в трьох вимірах, використовуючи камеру та світло, випущені з палички, але на відміну від Kinect, гравець повинен тримати паличку в руці для тривимірного відстеження.

Xbox Kinect був випущений в листопаді 2010 року компанією Майкрософт для використання з консоллю Xbox 360. З його випуску і до березня 2011 року було продано понад 10 мільйонів одиниць, це було найшвидшим продажом електронного приладу взагалі [33]. Kinect має функції контролю руху, голосових команд і розпізнавання обличчя, і є прямим конкурентом Nintendo Wii і Playstation Move. Що є унікальним в Kinect, це можливість керувати вашими відеоіграми лише своїм тілом без необхідності керувати рукою. Kinect складається з VGA-камери, датчика глибини та масиву мікрофонів. Він також має функцію моторизованого нахилу, яка дозволяє камері знайти ідеальний кут для найкращого знімка на сцені. Технічні характеристики Kinect також наведено в розділі 2.8

1.2 Робототехніка, як наука

Ідея автоматизованих машин живе тисячі років, перші сліди робототехнічних технологій можна знайти ще в 2000 рр. до н. е., коли єгиптяни робили автоматичні іграшки та пристрої для розваг. Пізніше, приблизно від 350 до 100 рр. до н. е., грецькі математики та винахідники створили автоматичних птах, дверні замки та годинники.

У 1350 році на вершині церкви в Страсбурзі був встановлений механічний півень, який керувався годинним механізмом і щодня махав крилами в полудень. У 1495 році Леонардо Да Вінчі створив робота гуманоїда, який виглядав як броньований лицар, механічний дизайн якого був зроблений так, щоб виглядати як справжня людина всередині робота.

З 18 століття нові розробки та машини з'являлися досить регулярно, більшість з них – моделі людей або тварин, які виконували прості рухи або могли грати музику, писати або малювати. У 1745 році британський винахідник Едмунд Лі продемонстрував використання зворотного зв'язку, коли він зробив вітряк, який завжди був зосереджений проти вітру. Один цікавий "винахід" з'явився у 1769 році, коли барон Вольфганг фон

Кемпелен побудував автоматизованого шахіста "Тюрк" – механічний чоловік, який міг грати в шахи. Машина була хорошим шахістом і змусила більшість людей повірити в те, що це була справжня машина з штучним інтелектом, коли насправді її керував короткий чоловік у костюмі робота. Незважаючи на те, що машина була підробкою, вона розпочала дискусію про можливість створення машин із штучним інтелектом.

У 1801 р. Французький винахідник Джозеф Жакард запровадив перфокарти, які, ймовірно, найбільш відомі як пристрої програмування для перших комп'ютерів. Він використовував перфокарти для керування автоматизованим ткацьким верстатом. Пізніше в ХІХ столітті з'явилося представлення того, що повинно було стати основою комп'ютера, коли Чарльз Беббідж у 1822 році створив і виготовляв прототипи "різницевої машини" та "аналітичної машини". У 1847 році Джордж Бул представив булеву алгебру, яка сьогодні є основою для сучасного цифрового комп'ютера, і, таким чином, управління роботом. У 1898 році з'явився, мабуть, перше представлення телевізійного робота, коли Нікола Тесла демонстрував дистанційне керування човном на електричній виставці в Медісон Сквер Гарден. "Пристрій Тесли був буквально народженням робототехніки, хоча він рідко визнається за це досягнення. ... На жаль, винахід набагато випереджав свій час, і ті, хто спостерігав це, не могли уявити собі його практичного застосування" [10].

Слово «Робот» було введено в 1921 році чеським письменником Карелом Чапеком у його п'єсі «Росумські універсальні роботи», де люди, подібні до машин, виготовлені для роботи справжніх людей. Слово "робот" походить від чеського слова "робота" і може бути перекладено на "примусову працю". Перша роботизована система провідний-ведений була розроблена у 1951 р. Раймоном Герцем для Комісії з атомної енергії. Рука була повністю механічною і використовувалася для обробки радіоактивних матеріалів, і вона розглядається як важлива віха у технології зворотного зв'язку [11]. У 1954 році Джордж Девол і Джо Енглебергер розробили першу програмувальну робочу руку, а в 1956 році вони почали працювати над першим промисловим роботом у світі, "The Unimate", який зайняв місце за конвейерною лінією на General Motors в 1961 році, і виконував лиття металів під тиском з машин і зварювання автомобільних частин [31, с.3]. У 1966 році почалася робота "Shakey" в Стенфордському дослідницькому інституті. Це був один з перших "інтелектуальних" роботів, який міг обміркувати те, що він зробив. "Shakey" міг запланувати свій шлях від А до В і міг пересуватися навколо перешкод. Він не мав практичного застосування в тому сенсі, що не міг виконувати ніякої роботи, але його створення було важливим для розробки штучних методів контролю, робототехнічного бачення та планування шляху [12].

Відомий (master-slave) робот Герца, "The Unimate" і "Shakey" є важливими віхами в

робототехнічному розвитку, оскільки їх можна розглядати як перший телекеруємиий робот, перший завчасно запрограмований робот і перший автономний робот.

Телекеруємиий робот, або телеробот – це робот, керований оператором на відстані. Роботи можуть відрізнятися в тому, наскільки їм потрібне керування. Деяким роботам треба повідомляти про кожен рух, який він повинен зробити, тоді як інші потребують лише невеликого керівництва для виконання своєї роботи. Останні являють собою поєднання телероботів і автономних роботів. Телороботи корисні, коли оператор хоче, щоб робот виконував завдання в неструктурованому середовищі, де неможливо запрограмувати роботу заздалегідь [13].

Важливим аспектом телекеруємих роботів є зворотній зв'язок з оператором, щоб він або вона могли знати, що сказати роботіві робити далі. Є різні типи датчиків, які можуть бути використані для надання інформації оператору, але найважливішим джерелом зворотного зв'язку, ймовірно, є бачення. Існує два основних способи отримання візуального зворотного зв'язку. Перше і найпростіше вирішення полягає в тому, щоб оператор мав візуальний контакт з роботом, щоб він міг бачити, що робить робот у будь-який час. Цей варіант може працювати за певних обставин, але в багатьох випадках оператор не має можливості бути в тому ж місці, що і робот. Одним з варіантів вирішення цієї проблеми є розміщення на роботі свого роду камери, що надає оператору живий візуальний зворотний зв'язок.

Заздалегідь запрограмовані роботи описують роботів, які запрограмовані для виконання одного завдання, яке буде виконуватися від моменту коли ви запускаєте робота, і доки ви його не вимкнете. Це типові промислові роботи, які повторюють однакові завдання знову і знову. Цей вид робота дуже корисний у виробничому середовищі, де він може робити одне й те ж саме завдання більш точним та швидшим, ніж його зробить людина.

Автономні роботи – це роботи, які можуть виконувати різноманітні завдання в неструктурованому середовищі без постійного керівництва оператора [14]. Автономних роботів можна зустріти у багатьох різних середовищах, починаючи з вашого саду, виконуючи стрижку газону, або на Марсі, відкриваючи нову територію. Загальні особливості автономних роботів полягають в тому, що вони можуть взаємодіяти з їх середовищем і приймати рішення самостійно, без зовнішнього впливу. Вони роблять це, приймаючи дані від різних типів датчиків, таких як фотоапарати, датчики температури або сенсорів. Автономні роботи можуть використовувати штучний інтелект для вирішення і вивчення того, що вони повинні робити в різних ситуаціях.

1.3 Оцінка впливу ігрових технологій на робототехніку

Основною подібністю між ігровою технологією та робототехнікою є те, що вони залежать від комп'ютерних технологій. Коли комп'ютери стали кращими, з'явилися відеоігри та роботи. Так важко знайти якийсь конкретну думку, яка стверджує "Відеоігри зробили це, і тому роботи тепер можуть це зробити", оскільки вони обидва залежать від розвитку комп'ютера. Але є два моменти, де робото-розробка може скористатися ігровою технологією.

Моделювання є важливим аспектом у розвитку робототехніки, особливо для мобільних автономних роботів. З гарним моделюванням більшість розробок можуть бути зроблені в стимуляційному середовищі, що зменшує витрати в апаратній потребі та прототипах. Гарне моделювання є також важливою частиною відеоігор, оскільки користувачі хочуть, щоб їх ігри були максимально реалістичним. У [21] вони описують, як використовувати симулятор на основі технології, призначеної для розробки комп'ютерних ігор для створення моделювання середовища для свого робота.

Відеоігри також можуть бути використані для підготовки персоналу, як використовувати робот. У армії США, використовується " The Robot Vehicle Trainer", відеогра, що імітує бойові середовища та використовує ті ж самі елементи керування, що і справжній робот. Ця гра використовується для навчання солдатів у використанні робота для демонтажу бомб [15].

Ігрові консолі часто мають багато можливостей обробки, якими скористалися ВПС США. Вони побудували суперкомп'ютер об'єднуючи 1760 консолей Playstation 3. Ігрові консолі часто набагато дешевше найкращих сучасних комп'ютерних частин і в цьому випадку вони використовуються лише на 5-10% від того, що подібна система коштувала б з дешевих комп'ютерних частин. Суперкомп'ютер, серед іншого, використовується для моделювання дослідження розпізнавання та штучного інтелекту [16], обидва є важливими речами в робототехніці. Звичайно, ідея обв'язування 1000 Playstations до робота явно неможлива, але це був би досить потужний робот, якщо б ви могли мати бездротовий зв'язок між роботом та комп'ютером.

1.4 Приклади використання робототехніки в медицині

У медицині перші роботи були введені приблизно в 1985 році, коли і використовувалися в нейрохірургії [40]. У нейрохірургії, яка вимагає крайньої точності, робот можна використовувати для розміщення інструментів та свердління отвору через череп і в мозок. Робот менш інвазивний, ніж людський оператор і може слідувати траєкторіям, недоступним людям. Роботи можуть працювати на різному рівні автономії, від тільки розміщення свердла, що використовується для буріння кісток, для розміщення свердла, свердлення отворів та потім розміщення хірургічного інструменту. Інше раннє застосування для роботів було в ортопедії, де працює робот високої точності використовувалася для свердління порожнин у кістці перед протезуванням. Першою системою для цього був Robodoc [30], якого було використано на тисячах хворих [40].

Ці системи були в основному використані на твердих недеформованих конструкціях, що дозволило заздалегідь спланувати весь рух. У 1990-х роках фокус повернувся на розробку більш інтерактивних робототехнічних додатків, що могли б використовуватись у ситуаціях, коли неможливо спланувати всі траєкторії попереду операції. Це призводить до телепередачі роботів з використання підрозділів ведучого та відомого. Існують різні методи контролю робота в таких системах, одним із способів є використання ведучого та відомого пристроїв. В таких системах, основний блок контролює оператор в той час як підпорядкований блок імітує рухи, зроблені на головному пристрої. Відомий може безпосередньо копіювати рух ведучого або налаштовувати рухи на зменшення швидкості та сили, або видалення невеликих ненавмисних рухів.

Останні тенденції в медичній робототехніці була розробка менших, більше легких роботів. Ці роботи можуть бути розміщені або приєднані до пацієнта [40]. Менше роботів можна простіше інтегрувати в роботу і призначити для виконання конкретного завдання. TEP [43] є прикладом такого робота на тілі системи. Роботи досить маленькі, щоб бути вставлені в тіло пацієнта. Ці системи зможуть рухатися безпосередньо на поверхні органів [40].

1.5 Технологія 3D-зображень.

Є дві загальних метода отримання інформації про 3D-сцену, пасивне та активне поглиблення. Пасивна глибина зондування використовує просіювання знімків з двох різних кутів для розрахунку глибини. Активні методи проєктують деяке світло на сцені, і використовують це для вимірювання відстані [28]. Нижче наведено коротке введення до трьох різних поглиблених відображень техніки: триангуляція, структуроване світло та час польоту.

Триангуляція може бути або активною, або пасивною [24]. У активній формі, промінь світла, як правило, лазер, проєктується на сцену в полі зору. Потім камера фіксує зображення сцени та положення світла зареєстрованого на зображенні. Відстань, D , до об'єкту точка лазеру проєктується на:

$$D = \frac{B}{\frac{x_0}{f} + \tan\alpha} \quad (1.2)$$

де B - вихідна відстань між лазером та центром оптичної камери,

x_0 - позиція лазерної точки на зображенні,

f – фокусна відстань камери

α - кут між проєктованим лазером і оптична віссю камери

Пасивна форма триангуляції полягає в тому, щоб використовувати декілька камер для прийому зображення сцени, це також відомо як стереосенс. У стереозвучі відстань обчислюється шляхом триангуляції між положеннями камери і відповідних пікселів на знімках.

Структуроване світло - це ще одна форма активної триангуляції. У структурованому світлі на сцену запроектований відомий рисунок, розраховується глибина на основі триангуляції між відомим еталонним шаблоном і відстеженим відбитим образом [27].

Час польоту, TOF - це метод вимірювання глибини за допомогою джерела світла проєктованого на сцену. Потім глибина розраховується з часу затримки між світловим імпульсом, що випромінюється, а датчик виявляє відображене світло. Принцип дуже простий, а відстань D визначається як

$$D = \frac{c \cdot \delta t}{2}, \quad (1.2)$$

де c - постійна швидкість світла,

δt - виміряний час між світлом, що випромінювалось та виявленим відбитком.

$c \cdot \delta t$ ділиться на 2, так як відстань проходиться двічі, від випромінювача до сцени та від сцени до датчика [26].

Інший варіант TOF випромінює постійний промінь світла з відомим періодом і амплітудою. Відстань розраховується шляхом вимірювання фази затримка відбитого сигналу, порівняного з відомим опорним сигналом [26].

Візуальний контроль роботів - це головне поле в робототехніці. Візуальне сприйняття - це одне з найважливіших почуттів і, дозволяючи роботам бачити, створює великий потенціал для них.

Існує, в основному, два різних підходи, два візуальних робота управління: позиційне керування візуальним сервоприводом та візуальним керуванням на основі зображення [37, р. 407]. У позиційному контролі візуальний вхід використовується для пошуку 3D координати позиції, які можна використовувати для керування роботом. Проблема з позиційним керуванням - це генерувати ці координати в режимі реального часу.

Контроль зображення використовує зображення для прямого управління роботом. Нормальним методом є визначення деякої функції помилки, яка ґрунтується на функціях вхідне зображення, обчислює команди руху роботу.

1.6 Дослідження сенсору комп'ютерного зору Kinect

Kinect складається з інфрачервоного (ІЧ), джерела світла, датчика ІЧ-світла, RGB камери, мікрофонного масиву з чотирма мікрофонами, мотор для нахилу датчик і акселерометр для виявлення кута Kinect відносно горизонту.



Рисунок 1.1 – Microsoft Kinect

Як RGB, так і ІЧ-камери мають роздільну здатність 640x480 пікселів (VGA) і можуть доставити потік зображень 30 кадрів в секунду. У Xbox камери використовуються для відстеження гравців, в той час як масив мікрофонів використовується для випуску голосових команд. Використання декількох мікрофонів Kinect може розташувати джерело звуку, і, отже, він може знати, які з них гравці, що видали команду. Рисунок 1.1 показує Kinect і розташування різних частин. Для датчика глибини, ІЧ-випромінювач розташований ліворуч, а інфрачервоний датчик знаходиться праворуч. Kinect використовує лазер класу А щоб випромінювати інфрачервоне світло, і тому немає ризику пошкодження очей із застосуванням Kinect. "Мозок" в Kinect походить від системи PrimeSense PS1080 на чіпі [17]. PS1080 контролює ІЧ-випромінювач і отримує від нього інформацію з камери та мікрофона. Карта глибини розраховується на чіпі і всі вхідні датчики синхронізуються і надсилаються на головний комп'ютер через інтерфейс USB 2.0 [32] 1. Kinect використовує більше енергії, ніж USB з'єднання може доставити, і, отже, він потребує додаткового джерела живлення. Це енергозабезпечення в більшості випадків включається, коли ви купуєте Kinect, оскільки тільки новий Xbox 360 може забезпечити достатню потужність для Kinect.

Kinect має роздільну здатність 640x480 пікселів на глибині та RGB камері. Поле зору для датчика глибини становить 57° в горизонтальному та 43° в вертикальному напрямку. Інфрачервоне світло, що випускається з Kinect, спотворено через фільтр, тим самим даючи "випадкове" розміщення пікселів глибини. Якщо ми припускаємо, що піксель симетрично спотворений, то графіки на рисунку 1.2 на наступній сторінці

показано, як поля перегляду та дозволу змінюється з відстанню Kinect до об'єктів. Графік на рисунку 1.2а показує, як поле зору збільшується лінійно з відстанню. Рівняння (1.3) показує обчислення поля огляду.

$$Field\ of\ view = 2 * \tan\frac{\theta}{2} * distance, \quad (1.3)$$

Рівняння (1.4) показує як зменшується роздільна здатність у напрямку x і у як дистанція збільшується, як показано в 1.2b, тут α становить 640 для горизонтальних і 480 для вертикальної області перегляду.

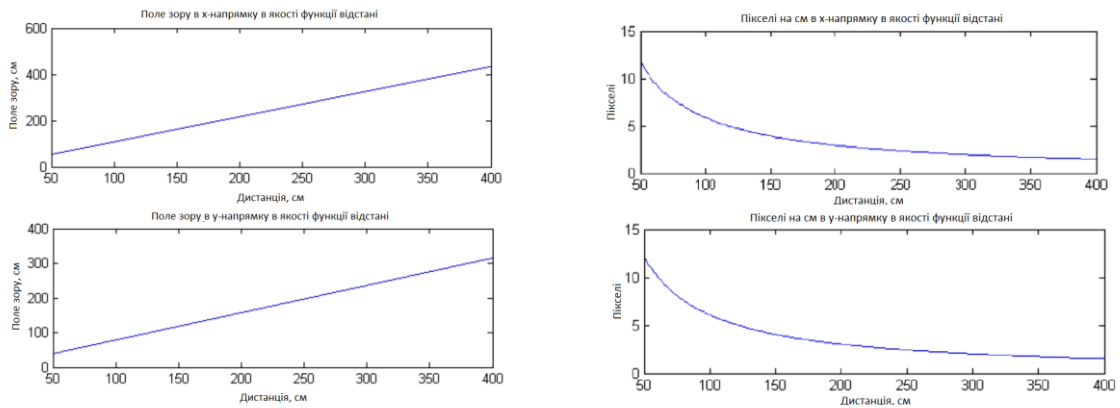
$$Pixels\ per\ cm = \frac{\alpha}{Field\ of\ view}, \quad (1.4)$$

На рисунку 1.2 ми бачимо, що зі збільшенням відстані роздільна здатність швидко зменшується, щоб досягти теоретичного дозволу 1 мм в x та у-напрямку, вимірювання повинні бути зроблені на відстанях нижче 60см. Microsoft повідомляє, що Kinect має відтворений діапазон 1,2 м - 3,5 м [29], в той час як PrimeSense повідомляє про те, що їх датчик використовує ту ж саму технологію, що може працювати на рівні 0,8 м - 3,5 м [38]. Під час тестів Kinect давав дані глибини від ≈ 50 см та [37] повідомлення про те, що вони є вимірюванням відстані до 15 м з Kinect. Так що ясно, що Kinect діє на більш широкому діапазоні, ніж повідомляють виробники.

Щоб виміряти глибину у своїй області перегляду, Kinect використовує структуроване світло, або Light Coding™, як PrimeSense називає це. ІЧ-світло з відомим рисунком викидається на сцену, а коли світло повертає спостережуваний рисунок використовується для розрахунку глибини. Це робиться всередині Kinect до порівняння спостережуваного шаблону із еталонним шаблоном, що зберігається в Kinect. На рисунку 1.3 показано, як виглядає ІЧ-шаблон з Kinect на плоску стіну Отримана точність дорівнює 1 см на відстані 2 м, і Kinect використовує 11 бітів для позначення значень глибини, що обмежують глибину роздільною здатністю до $2^{11} = 2048$ рівнів квантування.

Kinect був призначений як ігровий пристрій для використання з Xbox 360, але це не зайняло багато часу перед першими драйверами з відкритим кодом для підключення Kinect до комп'ютеру потрібен доступ до Інтернету. Багато програмістів бачили потенціал у цій новій технології і хотіли використати її. Це призвело до спільноти з відкритими кодами, які створюють та використовують код для використання з Kinect.

Сьогодні існує три основні джерела драйверів для Kinect. Першим джерелом драйверів є ті, які створено спільноту з відкритим кодом OpenKinect та їх бібліотека програмного забезпечення "libfreenect", яка забезпечує драйвери що дає вам прямий доступ до вихідних даних від Kinect.



(а) Поле зору

(б) Роздільна здатність на см

Рисунок 1.2 – Розподільна здатність і поле зору Kinect

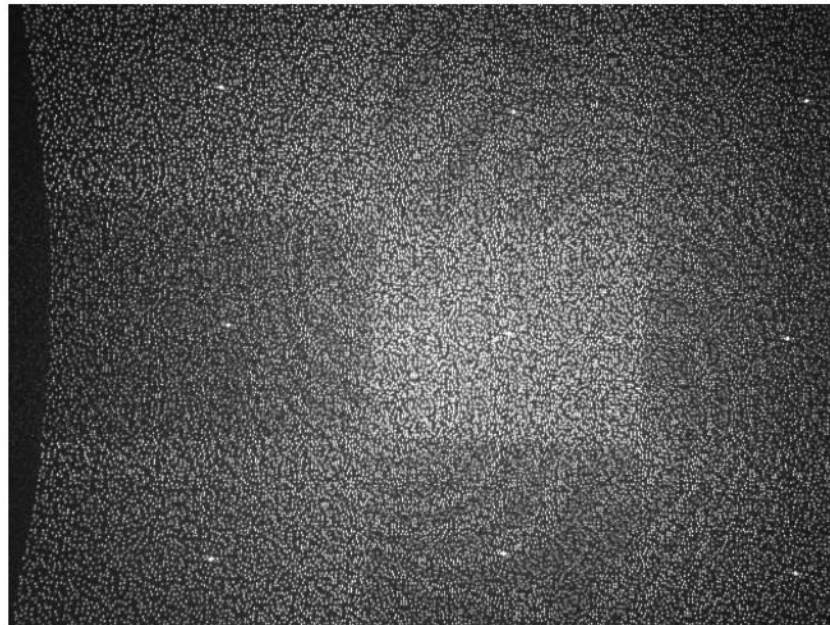


Рисунок 1.3 – Kinect ІЧ-шаблон, зображення з [1]

Іншим альтернативним програмним забезпеченням є програмне забезпечення, надане OpenNI, яка є некомерційною неприбутковою організацією, що працює в галузі природних технологій взаємодії людей і машин. Один з основних вкладників у OpenNI - це PrimeSense™, компанія, яка використовувала 3D-технології в Kinect.

Останньою альтернативою є використання офіційних драйверів, випущених

Microsoft. Ці драйвери були випущені разом із комплектом програмного забезпечення для розробки програмного забезпечення SDK для Kinect в травні-червні 2011 року, а також оновлено, коли компанія Microsoft випустила Kinect для Windows у 2012 році.

Бібліотека Point Cloud [34], PCL, є бібліотекою з відкритим кодом для обробки 3D точкових хмар. Він містить кілька алгоритмів обробки точкових хмар, таких як фільтрація, вилучення функції та вирівнювання точкової хмари. PCL також має обгортки, що надають прямий доступ до бібліотеки OpenNI, тож з PCL можна отримати доступ до всіх тих самих функцій, що і у OpenNI.

Таблиця 1.1 – OpenNI проти Kinect SDK

Kinect SDK	OpenNI
Пряма підтримка управління двигуном	Немає підтримки керування двигуном
Підтримка аудіокоманд	Немає підтримки аудіо
Пропозиція повного відстеження тіла без калібрувальна позиції	Можливість відстеження тіла з калібруванням позиції
Тільки повне відстеження тіла	Підтримка ручного відстеження та вбудоване розпізнавання жестів
Тільки в Windows 7	Платформо незалежна
Може бути використаний тільки з Kinect	Може бути використаний з Kinect та Asus Xtion Pro
Лише некомерційна ліцензія	Ліцензія з відкритим кодом

OpenNI і PCL визначають джерело координат системи Kinect, щоб бути в центрі датчика глибини. Вісь z спрямована прямо з датчик, вісь x ліворуч, а вісь y спрямована прямо вниз.

Фреймворк OpenNI забезпечує передавання карту глибини, яка складається з 640 * 480 вимірювань глибини. Кожна точка на карті глибини являє собою відстань від датчика глибини Kinect до об'єкта(-ів) у місці видимості. Глибина значення являє собою відстань від площини, розташованої на початку датчика, перпендикулярно до осі z датчика глибини, а також до об'єктів зору. Іншими словами, відстань до кожної точки на плоскій поверхні перпендикулярно до осі z Kinect є однаковою. OpenNI забезпечує значення глибини в міліметрах, а PCL перетворює його в метри. PCL використовує глибину показання для автоматичного обчислення і створення точкової хмари, де кожна точка представлена координатами XYZ. Також можна отримати дані разом з датчиком глибини, вирівняним з зображенням з RGB камера. Це надасть зображення RGB-D з кольором та

глибиною інформація зі сцени.

Існують доступні приводи, що можуть бути використані для доступу до двигуна Kinect з програм OpenNI. Цей привід вимагає додаткового встановлення, якщо потрібно отримати доступ до акселерометру Kinect. Тому SDK від Windows часто використовується для керування двигуном Kinect.

Комп'ютерне бачення є важливим аспектом у робототехнічних системах. З Kinect можна придбати майже повну систему машинного бачення за трохи грошей. Це відкриває велику кількість нових досліджень та розробок, адже тепер можна створювати передові візуальні системи без значних фінансових ресурсів. Там було розроблено безліч додатків, що використовують Kinect. Дуже багато з них не дуже корисні, але деякі з них дійсно показують діапазон того, що можна зробити з Kinect. Нижче наведено три приклади того, що Kinect використовується сьогодні.

Роботи були використані в операційній кімнаті протягом багатьох років, однією з основних проблем робототехнічної хірургії є відсутність фізичних відгуків оператору. Група студентів з Університету Вашингтона знайшла можливе рішення для цього [36]. Використовуючи Kinect, вони зробили програму, яка відображає 3D-середовище робота, і визначає області, з якими робот не повинен залишатися. Коли робот збирається в'їжджати в незаконну зону, то джойстик, який використовує оператор для керування, робот зупиниться. Це говорить оператору, що він збирається зробити щось, чого він не повинен робити. Інша велика річ про цю систему, наскільки це дешево, аналогічна система без Kinect буде коштувати близько 50 000 доларів, тоді як ви можете придбати Kinect для близько 150 доларів.

Інше використання Kinect в операційній кімнаті не має нічого спільного з роботом управління, але з маніпулюванням зображень. У хірургії образи допомагають хірургу знайти свій шлях у організмі пацієнта. Але якщо хірургу потрібно знову поглянути на картинку, може бути складно пояснити іншому, яку фотографію він хоче і де саме він хоче збільшити. Це стає ще гірше, якщо хірург повинен йти на комп'ютер і маніпулювати зображеннями сам, то йому доведеться залишити стерильне середовище на операційному столі, а потім знову прибрати, коли він повернеться. Це займе багато часу при операції. Тепер це рішення було зроблено, використовуючи Kinect, тепер можна простежити рух хірурга в трьох вимірах. Таким чином, він може керувати фотографіями, які показуються на великому екрані в операційній кімнаті. За допомогою хвилі рук він може змінювати картинку, обертати їх і збільшувати і зменшувати, заощаджуючи багато часу для хірурга [22]. Контроль зображення виконується так само, як на сенсорному екрані клавіатури або телефону, крім рухів робиться в середині повітря.

Kinect також використовується для отримання візуального введення автономних роботів. Це показано в [38, 25], де Kinect прикріплений до малих чотирикутних вертольотів, і дає квадрокоптерам тривимірний вигляд світу. Це дозволяє їздити автономно навколо, не стикаючись з іншими об'єктами.

1.7 Порівняння з іншими датчиками

З моменту його випуску, було зроблено кілька досліджень на Kinect та якості їх вимірювань. У [49] точність підсилувача Kinect порівнюється з Acerized Finder Range Finder, aLRF, який є популярним датчиком глибини для мобільних роботів. У цьому ж тесті Kinect також порівнюється з двома іншими невисокими витратами часу польоту, TOF, 3D-датчиками, SR-4000 та Fotonic B70. Їх тест показує, що Kinect дає кращі показники, ніж два TOF датчики, а діапазони до 3,5 м, Kinect може бути хорошим заміником aLRF, оскільки їх продуктивність аналогічна до цього діапазону. У [27] показано, що похибка вимірювання глибини зростає квадратично з відстанню. І вони заявляють, що для додатків для картування глибини дані повинні бути придбані в межах 1-3 метрів від датчика. На більших відстанях якість даних пошкоджена шумом з низьким дозволом. [41] порівнює Kinect зі стереозвищенням, досягнутим двома піксельними камерами з роздільною здатністю 3,5 мегапікселя, і камерою TOF. Продуктивність Kinect була близька до того, що постачається двома дзеркальними камерами, і набагато краще, ніж TOF. Вони також моделювали роздільну здатність Kinect в глибині, $q(z)$, щоб бути функцією від відстані z , див. рівняння (1.5).

$$q(z) = 2.73z^2 + 0.74z - 0.58 [mm], \quad (1.5)$$

[28] використовує глибинну камеру PrimeSense, побудовану на тій же технології, що і Kinect, для створення моделей об'єктів, що знаходяться в роботі. Вони показують, що камера дає досить хороші дані для відтворення 3D-моделей поверхонь об'єктів.

2 МЕТОДИ ОЦІНКИ ДАТЧИКА КОМП'ЮТЕРНОГО ЗОРУ KINECT

2.1 Методи тестування та налаштування

У робототехніці та особливо в медичних додатках дуже важливо, щоб обладнання, що використовується було надійним. Досить важливим є отримання високої точності accuracy та precision. Створений набір тестів для того, щоб більше дізнатись про Kinect'ові accuracy та precision. У цьому розділі наведено введення до всіх тестів виконаних на Kinect.

Було зроблено багато звітів про accuracy та precision Kinect і точність, але всі вони протестовані Kinect на нерухомих об'єктах. В додатку до тестування Kinect на нерухомих об'єктах, метод перевірки можливість відстеження Kinect для рухомого об'єкта наведена в цій главі.

Було виконано серію тестів, більшість з яких мають наступну структуру:

- ініціалізація Kinect;
- очікування вимірювання глибини падіння;
- буфер N кадрів;
- фільтрація області інтересу з буферних кадрів;
- зберігання даних у файл;
- експорт даних у MATLAB;
- обробка даних в MATLAB.

Крок 2 в тестовій структурі виходить з того, що після початку захоплення кадрів, йому потрібен деякий проміжок часу для калібрування перш ніж знаходить правильну глибину. Це видно на рисунку А.1 в додатку А. Здається, зміна початкової вимірюваної глибини після приблизно 40 кадрів, що приблизно становить 1,33 секунди. Отже, ці вимірювання змінювалися, уникаючи ручного запуску буферизації в кожному тесті після короткого періоду часу. У всіх тестах Kinect був розташований проти плоскої стіни або напроти якогось тестового об'єкту, як показано на рисунку 2.1 на наступній сторінці.

Перші тести призначені для того, щоб дізнатись більше про технічні деталі Kinect, якщо виміряна роздільна здатність відповідає вказаним специфікаціям. Існують певні тести для дослідження Kinect точність та accuracy. Після цього дані від Kinect використовуються для відтворення сфери, щоб з'ясувати, чи можуть дані Kinect бути використаними для оцінки об'єкта. Потім оцінка кута поверхні повинна мати відношення до Kinect, щоб бути зафіксованою датчиком глибини, а потім тестом на можливості

вимірювання положення рухомого об'єкта Kinect.

Оскільки Kinect є найбільш точним на близьких відстанях [27], а також роздільна здатність має експоненціальне збільшення з відстанню [37], більшість з цих випробування були прийнятими в межах 0,5 м - 1,5 м. Деякі випробування, коли доступний простір дозволяв це, були зроблені з діапазонами до 4 м.

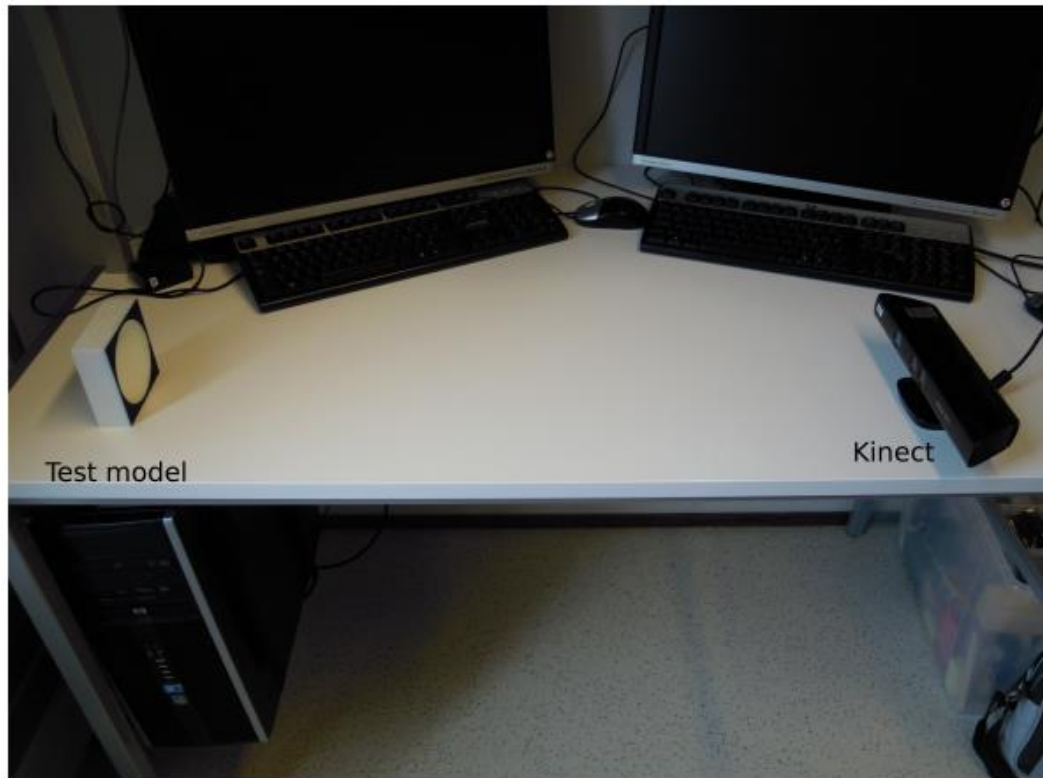


Рисунок 2.1 – Типова тестова установка Kinect

2.2 Методи тестування роздільної здатності

Як вже згадувалося раніше, роздільна здатність Kinect в діапазоні зменшується, коли відстань збільшується. Отже, щоб підтвердити [37] модель глибини роздільної здатності, аналогічний тест був виконаний. Kinect був поміщений перед плоскою стінкою і коли Kinect повільно віддалявся від стіни, вся глибина вимірюється зі збереженого центру пікселя Kinect. Далі дані були передані до MATLAB, де були вказані всі зареєстровані значення глибини були відсортовані та розраховується розмір кроку між кожною величиною глибини. Розмір кроку буде означати розмір кроку глибини Kinect. Щоб переконатися, що Kinect був переміщений досить повільно, щоб не пропускати жодних значень глибини тест проводився чотири рази, і всі дані з чотирьох тестів були

об'єднані.

Замість того, щоб зчитувати глибину лише з одного пікселя, можна скористатись середнім значення кількох пікселів для оцінки глибини. Це буде збільшуватися роздільну здатність глибини. Один із способів зробити це - визначити сітку в (x, y) просторі, а потім середні всі вимірювання глибини в цій сітці в нове одиночне значення глибини. Це також зменшить кількість даних точок, що буде дуже добре, якщо у вас не велика потужність обробки. Звичайно, це відбувається за рахунок нижчої роздільної здатності в (x, y) площині. Інша альтернатива, якщо не хочемо втрачати роздільну здатність, - це використання середнього фільтру на всі обрані дані. Тоді кількість точок буде однаковою, але їх значення глибини будуть створені з середньої глибини їх сусідів. Це рішення потребує більшої обчислювальної потужності, але буде зберігати щільність точок. Недоліком цього підходу є те, що глибина країв буде згладженою.

На основі цього методу було проведено друге випробування роздільної здатності, з використанням тих же налаштувань, що і раніше, але на цей раз розмір кроку глибини був розрахований за середньою глибиною від усіх пікселів в межах площі 1 cm^2 в центрі представлення (view) Kinect.

Роздільна здатність Kinect в x і y напрямках зменшується, оскільки збільшується відстань між Kinect та об'єктами. Це теоретично розрахована роздільна здатність через представленням (view) кута Kinect та передану роздільну здатність. Цей тест призначений для керування цими розрахунками. Kinect розміщувався перед плоскою стінкою та площею $20 \times 20 \text{ cm}$ у центрі представлення, що було відфільтроване. Тоді кількість вимірювань глибини виконана з цієї площі, що була підрахована. Це було зроблено за 100 кадрів, а потім Kinect був переміщений далі від стіни і ця процедура була повторена. Середня кількість точок даних, отриманих у цих 100 кадрах на кожній відстані була використана для порівняння вимірюваної роздільної здатності до розрахункової роздільної здатності.

2.3 Методи тестування стабільності і точності

Kinect продається як дешевий споживчий електронний прилад, тому якість в електроніці може коливатися. Для цього проекту важливо, щоб глибина вимірювань залишалася стабільною та надійною. Цей тест був виконаний, щоб з'ясувати більше як якість глибинних зображень перевершує на всьому полі представлення, якщо будь-яка

частина зображення краща за інші. Це було зроблено шляхом розміщення Kinect безпосередньо перед плоскою білою стіною, Kinect було поміщено досить близько до стіни, щоб в полі зору не було нічого окрім стіни. Тоді всі захоплені вимірювання глибини з Kinect зберігалися за 100 кадрів. У MATLAB ці дані були використані для створення двох зображень, одна глибина зображення, що містить середню глибину до стіни для всіх 100 кадрів, і одне зображення, зроблене шляхом підрахунку кількості дійсних вимірів глибини для кожного пікселя у всіх 100 кадрах. Зображення глибини використовувалося для перегляду якщо середня глибина до стіни була різною в різних регіонах зображення. Оскільки стіна вертикально плоска і перпендикулярна лінії зору Kinect, глибина повинна бути однаковою для всієї картини. Отримати вказівки про те, як змінюються вимірювання в глибині зображення, як були розраховані середня глибина і стандартне відхилення по горизонталі та вертикалі, починаючи з центру зображення та до верхнього та правого краю зображення. Зображення, що показує дійсну кількість вимірювань, яку можна використати, щоб побачити, де в області перегляду Kinect він втрачає більшість даних.

Сенс цього тесту полягає в тому, щоб дізнатись більше про точність Kinect різної довжини від поверхні. Точність - це здатність пред'являти ті ж результати знову і знову. Обчисливши середню глибину з вимірювань стандартне відхилення може бути використане для оцінки точності. Kinect стоїть навпроти плоскої стіни і виміри глибини, повернуті з датчика, були відфільтровані так, що тільки площа 20x20 см, з центром в x- і y-координатах (0, 0), була використана в подальшому розрахунку. Показання глибини від усіх пікселів у цій площі за 100 кадрів були збережені для подальшої обробки в MATLAB. За кожні 100 кадрів датчик трохи далі рухався від стіни. Усього було 7 тестів виконано, на відстані 56см до 200см. У MATLAB середня відстань і стандартне відхилення до стінки було розраховане з усіх показники глибини в кадрі. Аналогічний тест було зроблено в [27], і вони знайшли похибку, що збільшувалася з відстанню квадратично.

2.4 Accuracy

Accuracy - це здатність робити виміри, близькі до реального значень. Тут слідують два тести, призначені для того, щоб дізнатись більше про вимірювання Accuracy Kinect в глибині простору, а в 2D xy-просторі.

Оскільки роздільна здатність зменшується, коли збільшується відстань, очікується що точність буде зменшуватися і з відстанню. Але оскільки отримані результати є середніми вимірними довжинами від 100 кадрів, існує можливість, що Accuracy буде

залишатися приблизно однаковою, у той час поки precision має найбільше зниження. Це тому, що передбачається бути в рівній мірі ймовірно що виміряна відстань або надто довга, або надто коротка. І більше навіть ніж 100 кадрів можуть дати правильні середні довжини.

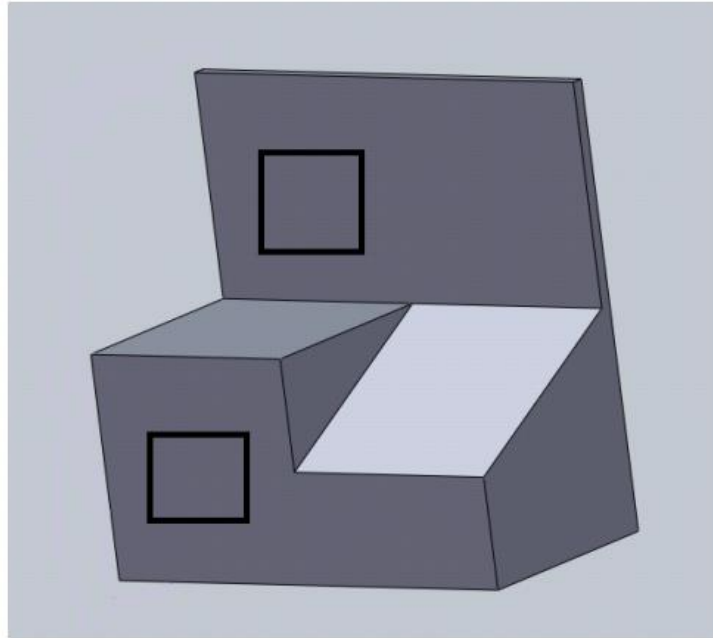


Рисунок 2.2 – Тестова модель, що використовується для тестування точності випробувань глибини та краю

Найважливішою особливістю Kinect є вимірювання глибини. Тому важливо знати більше про абсолютну точність глибини. Одним із очевидних способів вимірювання було б розміщення Kinect спереду поверхні, виміряти глибину з Kinect, а потім зробити інше вимірювання глибини з іншим пристроєм, а потім порівняти результати.

Це те, що зроблено в [39], де вони порівнюють Kinect з Actuated Laser Range Finder, aLRF. У цьому проекті єдиний інший доступний вимірювальний пристрій був столярною лінійкою, і спроби вимірювати цю ж відстань таким пристроєм виявилися головним джерелом помилок. Рішенням для цього, тестова модель (рисунок 2.2) була розроблена в Solidworks і надрукована в 3D. Відстань між найближчою поверхнею і найдалшою поверхнею на цій моделі рівно 5 см. Початковий план полягав у тому, щоб фільтр області 2x2 см тестової програми був дозволений на обох поверхнях, а потім - середню відстань до пікселів у цих областях може бути розрахована. Але коли відстань між Kinect та моделлю збільшилася стало важко отримати достатньо добре ручне вирівнювання Kinect і моделі, тому ці виміри є недостатньо точним для того, щоб бути корисним. Замість цього Kinect брали глибинні зчитування з усієї фігури і дані були передані в MATLAB, де

області інтересів були вручну розташовані та відфільтровані.

Ці регіони ілюструються чорними квадратами на рисунку 2.2. 100 кадрів даних глибини з Kinect були захоплені, і для кожного кадру відстань Δ_z була розрахована за рівнянням (2.1), де z_1 та z_2 - середня відстань до всіх пікселів в найдальших та найближчих квадратах.

$$\Delta_z = z_1 - z_2. \quad (2.1)$$

Вимірювана різниця глибини Δ_z , була розрахована для кожної з 100 кадрів на кожній відстані між датчиком та моделлю.

Тест «ХУ ассигасу» був зроблений, щоб дізнатися більше про те, наскільки добре Kinect міг би вимірювати відстані в x- і y-напрямах. Коли глибина зображення з Kinect була перевірена візуально, стало зрозуміло, що дуже багато шуму навколо гострих країв.

Тест проводився шляхом розміщення паперу формату А4 на вікні, а потім розміщення Kinect прямо перед документом, переконавшись у тому, що є центральним пунктом в глибині датчика зображення розташованого на папері. Починаючи з центрального пікселя горизонтальна та вертикальна лінія була намальована, поки вони не дійшли до кінця паперу. Відстані між цими кінцевими точками були використані для розрахунку ширини і висоти паперу формату А4 . Знайдені відстані були збережені, а також вимірювана відстань до центральної точки. Як і раніше, це було зроблено для 100 кадрів і відстань між Kinect і папером була збільшена для кожного випробування, від 0,5м до 4м.

Для тесту «Ассигасу країв» використовувалася та сама модель, що і в тесті на «Ассигасу глибини». Якщо дивитися прямо на передню частину цієї моделі, можна побачити один гострий край і нахилену поверхню з двома менш гострими краями. Ці краї можуть бути використані щоб порівняти, як Kinect виконує навколо країв і поверхонь різні кути. Модель була поміщена вручну безпосередньо перед Kinect, настільки перпендикулярно до z-осі Kinect, наскільки це можливо. Тоді сто послідовних кадрів, знятих з Kinect, були буферизовані, а область 20x20 в центрі представлення, яка містила модель, була відфільтрована і дані глибини були збережені для подальшої обробки в MATLAB.

У MATLAB збережені дані використовувались для створення зображення глибини, як показано на рисунку 2.3а на наступній сторінці. Зображення складається з середньої глибини із всіх 100 кадрів. На цьому зображенні гострі і нахилені краї були вручну

розташовані та відфільтровані, як показано на рисунках 2.3б та 2.3в, ці виділені області з полями, щоб уникнути перешкод від вертикальних країв.

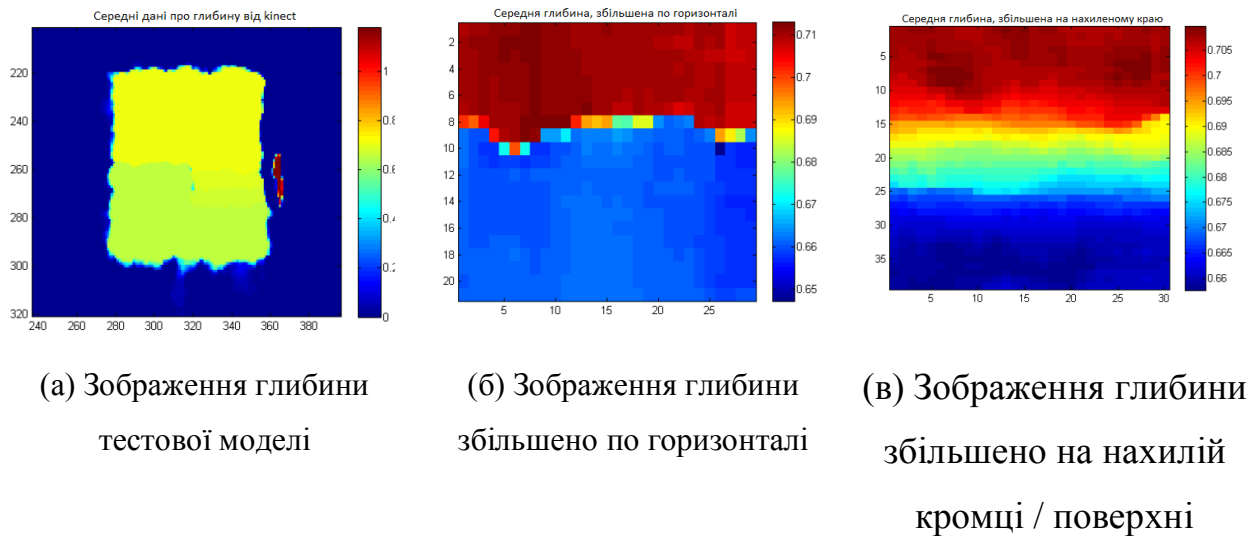


Рисунок 2.3 – Глибинний образ тестової моделі, що використовується для перевірки точності границі. Цифри на осі "y" - це лише піксельні індекси, додані MATLAB

Краї навколо моделі та гострий край посередині повинні бути прямими і краями, але, як ми чітко бачимо, це не так. На рисунку 2.3б ми бачимо різницю глибини між найближчими та найдальшими поверхнями, як червона і синя області. В ідеалі цей перехід між червоною та синьою областями повинен бути прямими лініями, але тут є три ряди пікселів, що вказують на край. Цей тест було зроблено протягом 8 циклів де відстань між Kinect і моделлю була збільшена з приблизно 10см рг. gun. Щоб більше сказати про асигуру краю, розраховано середнє значення глибини в піксельному рядку. Стандартне відхилення в кожному рядку може вказувати наскільки добре Kinect виявляє краї.

2.5 Модель реконструкції

Один із варіантів використання одиночних точок даних з Kinect для управління роботом – це дозволити зображенням з Kinect використовувати для створення моделі об'єкта прицілу. Тоді ця модель може бути використана для управління роботом у деякому роді. Цей тест призначений для з'ясування того, наскільки добре ми можемо відтворити модель з даними, отриманими від Kinect. Подібний тест був зроблений в [28], але вони використовували камеру PrimeSense замість Kinect. Їх тест показав хороші результати,

тому було цікаво подивитися, чи може Kinect зробити це так само.

Цей тест проводився на двох різних моделях. Перша модель була сферою, а друга модель була напівсферою. Зі сферою форму можна було виміряти з випуклої поверхні, а з половиною сфери дані були зафіксовані на випуклій стороні сфери.

Сфера з зовнішнім радіусом в 5 см була розміщена безпосередньо перед Kinect, позиціонування було зроблено так, що сфера була єдиним видимим об'єктом для Kinect в межах певного діапазону. Це полегшило фільтрування фону і знімання даних лише зі сфери. Кожну точку даних зафіксовану зі сфери, x-, y- та z-координати, було збережено. Для кожного пробігу було зафіксовано 100 кадрів даних, а тест було виконано на 10 різних відстанях між Kinect і сферою. У MATLAB збережені дані були використані для оцінки радіусу сфери та позиції сфери в системі координат Kinect. Ця оцінка була зроблена з варіанту методу найменших квадратів, знайдене у [18].

Окрім оцінки сферичної моделі, отримані дані були використані щоб зробити оцінку кута поверхні, що мають бути видимим для читання глибини відносно Kinect. Це пояснюється далі в розділі 2.7.

Цього разу при вимірюванні модель являє собою сферу увігнуту в сторону або в середину. Кривизна моделі складає 60% всередині половини сфери з внутрішнім радіусом 5 см. У тесті з повною сферою легко було помістити її так, щоб все інше було відфільтровано. За допомогою цієї напівсферичної моделі це було неможливо, головним чином через плоскі кадри навколо сфери. Щоб подолати цю проблему, плоский край був зафарбований чорним кольором, тоді як область, що цікавить (половина сфери) залишалася білою. Оскільки OpenNI / PCL має вбудовану функціональність для реєстрації зображення RGB з зображенням глибини, зображення RGB можна використовувати для визначення сфери і сегментувати правильні дані з вимірювань глибини.

Рисунок 2.4a демонструє RGBD зображення, зняті з Kinect. Використовуючи MATLAB, зображення RGB половини сфери було перетворено до двійкового зображення. Оскільки зображення складалося лише з площі 20x20 см в центрі представлення Kinect найбільшим об'єктом в очах була сфера, і тому кожен невеликий об'єкт (фонові об'єкти за моделлю і шум) можна легко видалити з зображення. Дані глибини Kinect часто складаються з великого шуму і, таким чином, залишаючи деякі отвори на поверхні сфери, ці отвори легко заповнювалися за допомогою `imclose` методу MATLAB. Зображення 2.4b показує коло сегментів з зображення RGB. Спостерігається невелике зміщення у вирівнюванні RGB і глибини зображення, щоб уникнути того, що частина тіла включена в знайдене коло, він трохи зігнувся, щоб гарантувати, що фон не буде включений в дані сфер. Тепер сферу можна відфільтрувати з даних глибини, використовуючи коло,

знайдене в RGB-зображенні як маску, рисунок 2.4с показує залишкові дані глибини після сегментації та маскуваннн. Як в попередньому тесті, центр сфери та радіус розраховували за допомогою методу найменших квадратів. Щоб отримати вимірювання, модель половини сфери була поміщена перед Kinect, з центром представлення Kinect приблизно в центрі сфери. Тоді всі вимірювання за межами центральної площини 20x20см, розташованої у центрі представлення, були відфільтровані. Для кожного налаштування дані RGB та глибини зберігалися на 100 кадрах. Тоді модель була переміщена приблизно на 10 см далі від Kinect для кожного випадку. Зображення 2.1 на початку цього розділу показує цей тест та налаштування.

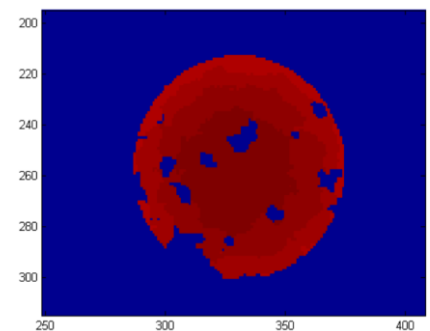
У наступному розділі представлені результати цих двох тестів. Там ми можемо побачити, що результати для даних, отриманих з увігнутої частини, у порівнянні з даними з опуклої сторони. Але, як зазначено за [27], а пізніше показано в тесті, описаному в 2.9.1, середовище навколо об'єкта може вплинути на вимірювання даних. Ці два випробування проводилися з двома різними налаштуваннями: повна сфера не мала поряд об'єктів, тоді як напівсфера стояла на рівному білому столі. Так ці екологічні відмінності можуть вплинути на результати і тим самим перешкодити порівнянню двох результатів. Отже, щоб отримати гарне порівняння, було зроблено третій тестовий запуск. Цього разу повна сфера була поміщена всередині половини сфери, щоб сфера могла бути сегментована від зображення однаково як півсфери.



(а) RGBD зображення з Kinect



(б) Сегментована половина сфери з RGBD зображення



(в) Глибинний образ відфільтрований із сегментованою сферою

Рисунок 2.4 – Кроки сегментування сфери з зображення RGBD Kinect

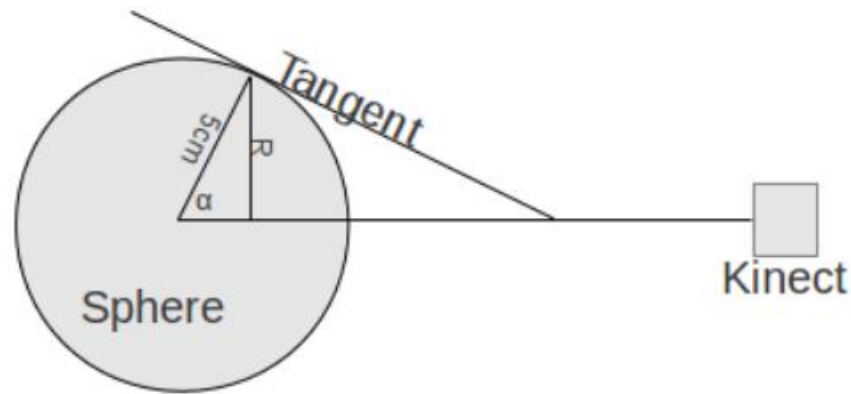


Рисунок 2.5 – Ескіз того, скільки сфери Kinect може зчитати даних глибини, і кут, якому це відповідає

2.7 Кут зору

На підставі даних попереднього тесту можна встановити оцінку кута поверхні об'єкта, який повинен мати відношення до Kinect, щоб бути видимим для Kinect. Сфера має радіус 5 см, тому, дивлячись на сферу в 2D видно круг із радіусом 5 см. В На рисунку 2.2с на ст. 19 ми побачили, як кількість пікселів на см кв. зменшилася, оскільки Kinect був переміщений далі від об'єкта. Підрахувавши кількість пікселів, які Kinect може прочитати з сфери, котру ми можемо встановити радіус кола, який відповідає цим вимірам. Від цього радіусу ми можемо зробити наближення максимального кута об'єкта, який може мати відношення до Kinect. На рисунку 2.5 показаний простий начерк розрахунку, де R - оціночний радіус. Ми зацікавлені в пошуку α , що являє собою кут між віссю Kinect і нормою дотичної з точок на сферу, де коло зі знайденим радіусом. α розраховується з:

$$\alpha = \sin^{-1}\left(\frac{R}{5}\right), \quad (2.2)$$

Коли α знайдено, можна визначити, що кут між віссю z Kinect і поверхнею нормалі об'єкту має залишатися в межах кута $\pm \alpha$ для Kinect, щоб мати змогу вимірювати глибину з поверхні.

Дані для цього обчислення беруться з першого тесту з повною сферою. У цьому тесті сфера була єдиним видимим об'єктом для Kinect, так всі точки захоплення даних можуть бути використані. У тесті, де сфера була сегментована на основі фонового кольору, деякі точки даних можуть бути втраченими, і, отже, оцінка радіусу була б неправильною.

2.8 Частотна характеристика Kinect та пропускна здатність

Частотна характеристика системи може розповісти нам про стаціонарні відносини в системі, а пропускна здатність говорить нам про оперативний діапазон системи. У [41, с.99-97] система та її відповідь, $y(t)$, на вхідний сигнал дає як показано у рівнянні (4.3). Вхід є синусоїдальною хвилею з амплітудою u_0 та частотою ω . $H(j\omega)$ - частотна характеристика, і $\varphi(\omega)$ – це затримка системи.

$$y(t) = u_0 |H(j\omega)| \sin(\omega t + \varphi(\omega)), \quad (2.3)$$

Чисельне значення частотної характеристики (4.4) - це співвідношення між амплітуда виходу y_0 та амплітуда входу u_0 . У розділі 7, де перевірена смуга пропускання робототехніки, u_0 , y_0 та $\varphi(\omega)$ показані на рисунку 7.1а.

$$|H(j\omega)| = \frac{y_0}{u_0}, \quad (2.4)$$

З [41, с.97] у нас є визначення смуги пропускання системи: частота характеристика зменшується, оскільки частота на вхідному сигналі зростає. Нарешті ми досягаємо вхідної частоти, яка не дає відповіді. Ця система називається фільтром низьких частот і спектром частот, де система реагує на вхідний сигнал, що називається пропускною здатністю системи. З [41] ми також отримуємо алгоритм для перевірки частотної характеристики системи та знаходимо пропускну здатність. Цей алгоритм показано нижче.

Алгоритм знаходження частотної характеристики системи:

- введіть синусоїду з частотою ω_i та амплітудою u_0 ;
- зачекайте поки вихід буде стабільним;
- виміряйте рівень виходу y_0 ;
- $|H(j\omega)| = \frac{y_0}{u_0}$;
- повторіть 1-4 від ω_0 до ω_n ;

Kinect працює на частоті 30 Гц, відповідно до теореми про вибірки, що вибірка повинна виконуватися з частотою щонайменше вдвічі вищої від початкової частоти сигналу, можна зробити припущення, що Kinect може відстежувати рух до 15 Гц. Рисунок 2.6 на наступній сторінці показує теоретичну оцінку реакції Kinect на рух зі збільшенням частоти від 1Гц - 15Гц. Оцінка здійснюється шляхом прийняття зразку на частоті 30 Гц із зростаючої частоти синусоїдальної хвилі, як описано в алгоритмі 1. Як правило, відповідь відображається в графіку передбачень, де величина виходу візуалізується разом із фазовою затримкою вихідного сигналу. Але тут зразки беруть безпосередньо з синуса, і тому це не затримка. Отже, для простоти відповідь побудована в "Нормальному" стилі, де ось x - вхідна частота, а ось y - вихідний коефіцієнт, див. рівняння (2.4). Частота зрізу в системі зазвичай визначається як у випадку, коли вихідна відповідь була зменшена в 3dB відносно вхідного сигналу. -3dB така ж, як $1 / \sqrt{2}$, тому в графіку позначено $1 / \sqrt{2}$. Оскільки частота вхідного сигналу збільшена, зразки вихідного сигналу стають квазіперіодичними для деяких частот, в цих випадках амплітуда виходу визначається як середня висота піків у вибіркового сигналі.

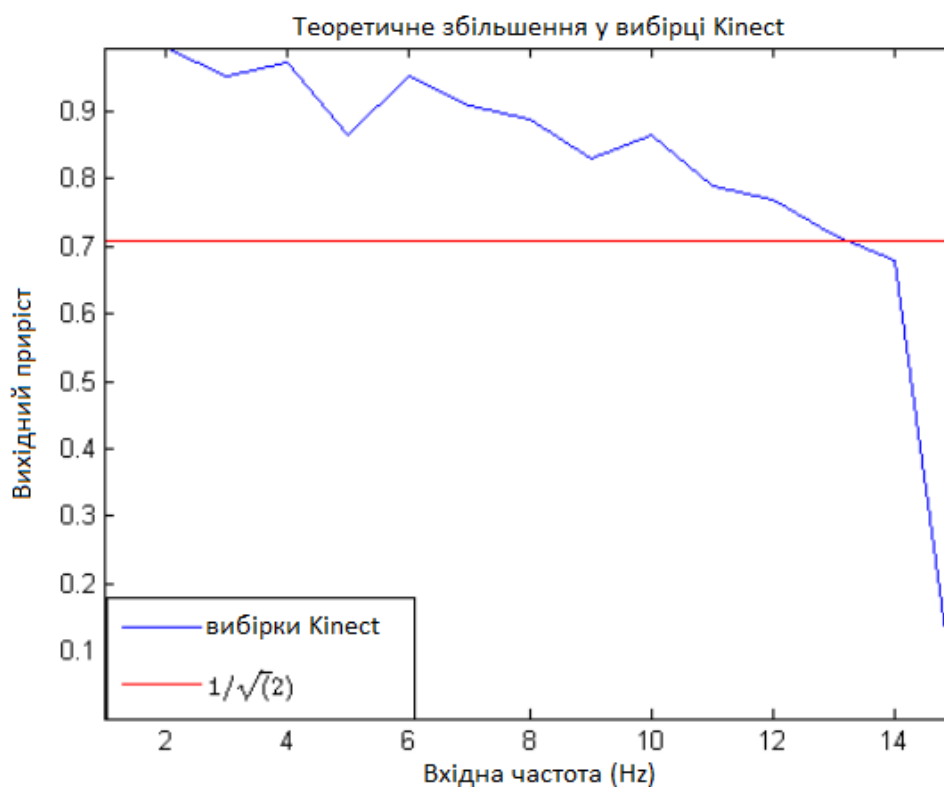


Рисунок 2.6 – Теоретичний приріст від відбору періодичних синусів з Kinect

3 РЕЗУЛЬТАТИ ТЕСТУВАННЯ KINECT

3.1 Роздільна здатність

На рисунку 3.1 викладено вимірюване значення глибини. Червона лінія на графіку - це роздільна здатність глибини, оцінена [37]. Виміряні дані виглядають відповідними прогнозованій розподільній здатності. Ми бачимо, що в найближчому операційному діапазоні 50см - 60см, є роздільна здатність 1мм. Від 60 см до 80 см, роздільна здатність переходить від 1 мм до 2 мм. Оскільки відстань виходить за межі 81см, найкраща роздільна здатність, яку можна сподіватися, становить 2мм. Можна також помітити, що при збільшенні відстаней інтервали до етапів квантування збільшуються.

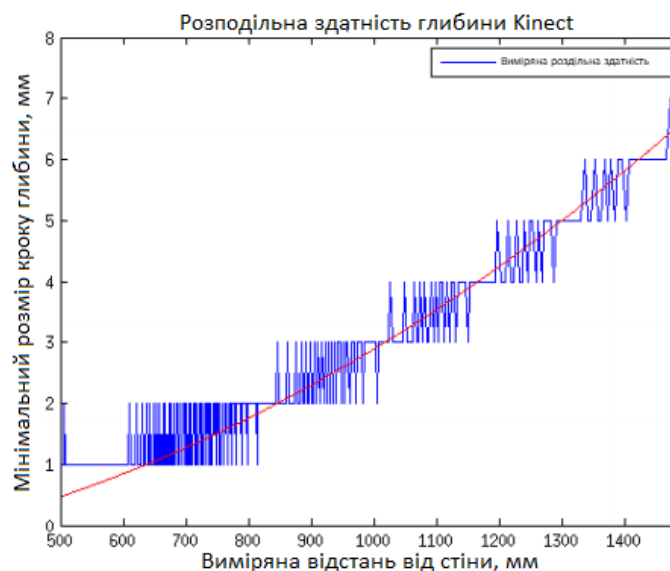


Рисунок 3.1 – Роздільна здатність Kinect в якості функції відстані. Червона лінія - це роздільна здатність, знайдена в [37], див. Рівняння (2.5) розділ 2.9

На рисунку 3.2 показано, як зменшується розмір кроку поглибленої глибини, якщо взяти середню глибину з 1 см², а не лише одного пікселя. Ми бачимо, що тепер ми можемо досягти роздільної здатності глибини менше 1мм на відстанях, що перевищують 1,5м, що є значним поліпшенням з роздільною здатністю 6 - 7мм на тій же відстані, використовуючи лише дані з одного пікселя.

На рисунку 3.3 на наступній сторінці показано кількість точок даних на см кв., зображених у квадраті 20x20см, на зразок теоретичного числа пікселів з рисунка 2.2с. Зразки приймаються в діапазоні 0,5 м-4м, і ми можемо бачити, що крива з вимірюваною роздільною здатністю добре слідує за теоретичною роздільною здатністю, але вона

постійно трохи нижче теоретичного максимуму. Це, мабуть, пов'язано з тим, що частина інфрачервоного світла не повертається до датчика, або що ІЧ-діапазон не є симетричним, як ми прийняли при розрахунку теоретичного дозволу.

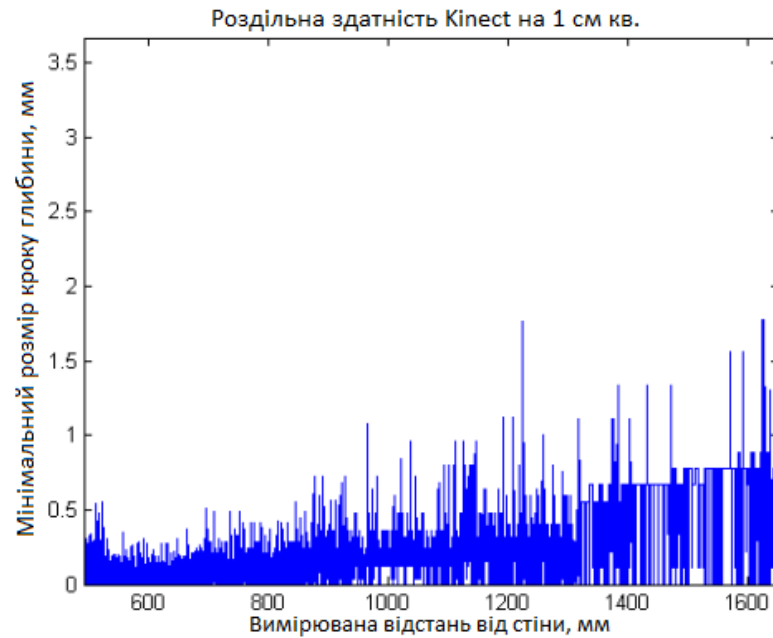


Рисунок 3.2 – Розміри Kinect в діапазоні від 1 см² від середньої глибини залежно від відстані

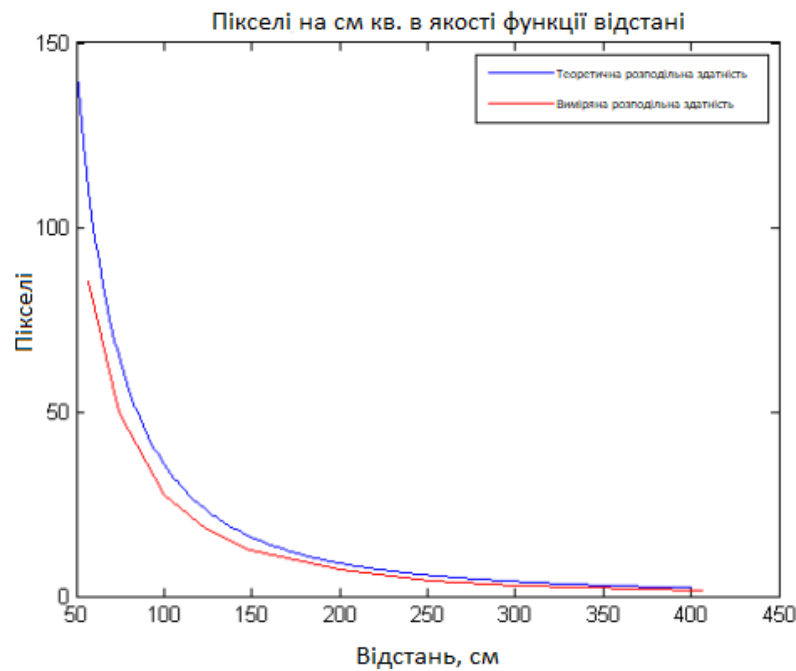


Рисунок 3.3 – Кількість пікселів на см кв. в квадраті 20x20см як функція відстані, побудована з урахуванням теоретичного дозволу

3.2 Стабільність

Графіки на рисунку 3.4 на облицювальній сторінці показують, як стандартне відхилення та середня виміряна глибина до квадрата на стіні змінювалися з часом. Червона лінія на графіках є середньою величиною, доданою для того, щоб продемонструвати коливання результатів.

На рисунках 3.4a та 3.4b наведені результати від 10 і 20 хвилин. Ми бачимо, що в середній глибині невелика різниця, амплітуда близько 1 мм. У цих двох перших графіках варіація незначна, і вона може здаватися навіть періодичною. Але якщо ми подивимось на рисунок 3.4c, який вимірює кожну секунду протягом години, ми можемо побачити, що середня вимірювана глибина збільшується більше ніж на 3 мм від початку до кінця. Здається, що глибина вирівнюється приблизно через 30 хв, але в міру того, як проходить час, є більше випадкових шумів (раптових крапель). Важко сказати, що це збільшення відстані, але це може бути пов'язано з варіаціями температури в Kinect. Тести проводились безпосередньо один за одним, в порядку, показаному на рисунку. Коли Kinect активний, він виробляє багато тепла, тому 10-хвилинний тест був з "холодним" Kinect, а останній з "теплим" Kinect. Цілком можливо, що зміна температури може трохи вплинути на довжину хвилі ІЧ та порушити показники глибини з часом. Щоб дізнатись, чи це кореляція між температурою та зміною вимірювань глибини, це може бути непоганим способом повторно запустити цей тест і контролювати температуру Kinect при тестуванні. Стандартне відхилення залишається стабільним за допомогою тестових циклів, за винятком кадрів, де середня глибина має раптові краплі. Причина, по якій 60-хвилинний запуск починається з меншого значення, ніж інші два цикли, полягає в тому, що Kinect було перенесено між секундами та третьою версією, а потім трохи втрачено під час відпуску.

На рисунку 3.5 ми можемо бачити зображення, що показують 100 кадрів від Kinect, де підраховується кількість недійсних вимірювань на кожному пікселі та формується зображення. Рисунок 3.5a показує 100 кадрів проти плоскої стінки, а Kinect - приблизно 54 см від стіни. Смужка праворуч від зображення вказує кількість недійсних пікселів, показуючи, що темно-сині значення 0 відсутні, а темно-червоний - означає відсутність усіх 100 значень. Що цікаво, цей результат - це кадр з недостатньою кількістю вимірювань, який проходить навколо 3 з 4 країв зображення. Kinect повинен доставити 640x480 вимірювань глибини, але чомусь дуже багато

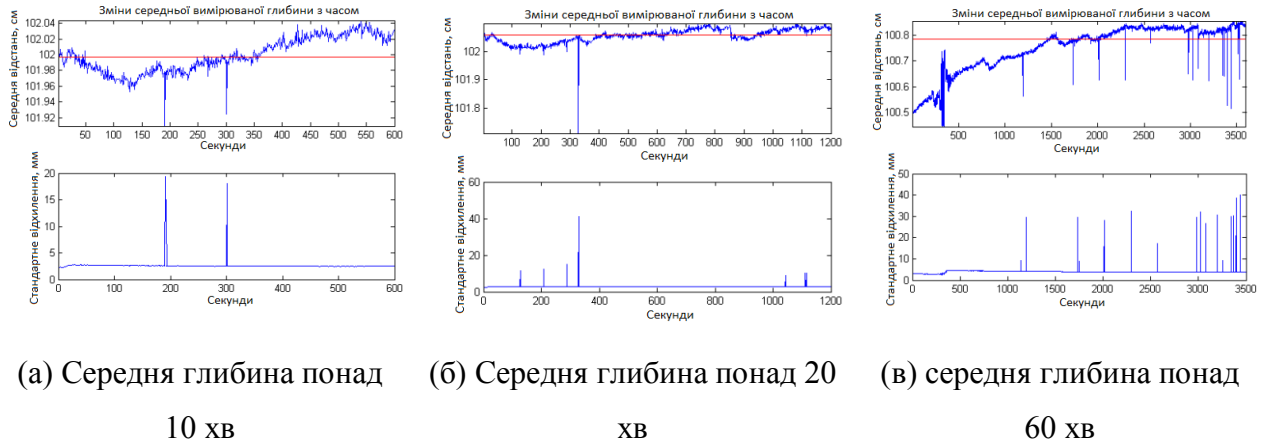


Рисунок 3.4 – Середня відстань між Кінест і стіною за часом

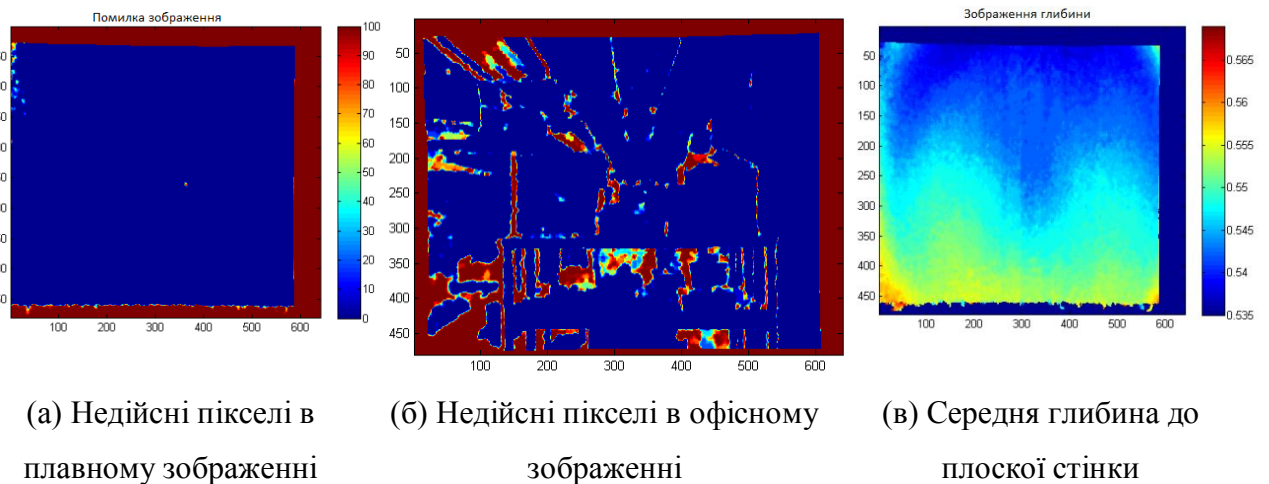


Рисунок 3.5 – Зображення недійсних пікселів

Значення відсутні навколо країв. Існує також невелика площа майже в центрі, де є багато відсутніх значень. Ці відсутні пікселі центру можуть потрапляти з бруду на об'єктивах ІЧ-камери або випромінювача, це може бути щось на стіні, що перешкоджає відбитці ІЧ-спектром, або це може бути помилкою в CMOS-чипі в ІК-камері всередині Kinect. Можливо, навіть Kinect розташовано, щоб закрити його до стіни, і це заважає результатам. Відзначено, що коли Kinect наближається до об'єкта, це перше значення втрачених пікселів центру.

На зображенні на рисунку 3.5b Kinect дивиться на офіс, тоді як кількість відсутніх значень пікселів розраховується на 100 кадрів. Тут є набагато більше пропущених значень на сцені, але це пов'язано з різним розміщенням і відстані до об'єктів зору. Деякі значення втрачаються через тіні, деякі повинні залишити датчик, щоб зареєструвати їх, тоді як інший не відображає ІЧ-світло назад датчику. Цікаво помітити на цьому зображенні, що кадр недійсних пікселів навколо зображення зараз має чотири краю, а кадр виглядає тонше чи зміщеним вправо. Одна теорія про те, чому з'являються ці краї, полягає в тому,

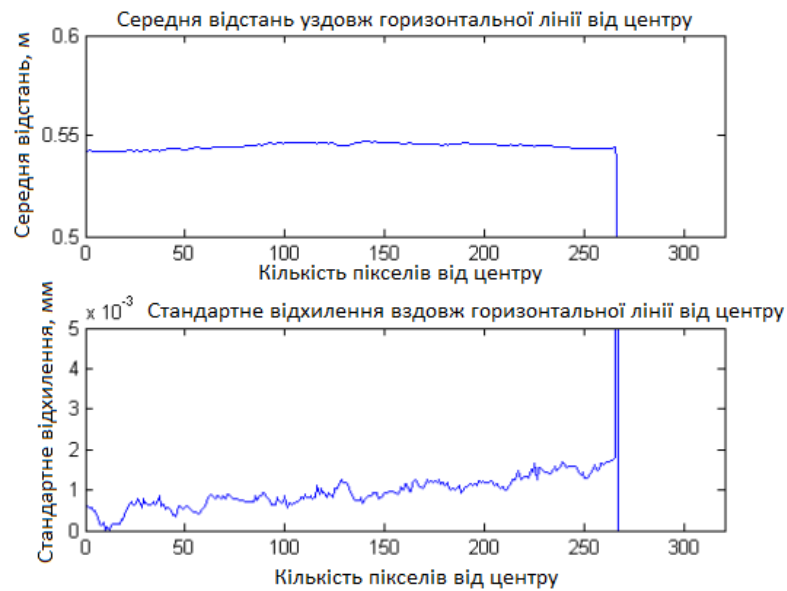
що у краях поля Kinect, інфрачервоне випромінювання, яке випускається з Kinect, може знаходитись в куті, який не дає достатньо віддзеркалення назад на Kinect. Kinect позначає значення як погані значення, якщо не впевнений, що може бути реальним результатом, отже, ми можемо отримати багато недоступних значень у краях. Але це не пояснює, чому відсутні пікселі складають близько 3 країв у першому випадку, а 4 - в останньому випадку. Але цей край відсутніх пікселів вказує на те, що розрахунок у полі Kinect, який був зроблений в розділі 2.8.2, є трохи помилковим. Фактична область перегляду трохи менше, ніж очікувалося. Це не велика проблема, але добре знати, що це трапляється.

Також можна зазначити, що місце, де відсутні значення пікселів у центрі зображення 3.5а плоскої стінки, відсутні на зображенні офісної сцени. Це може свідчити про те, що місце з'явилося через те, що Kinect було встановлено наблизитись до стіни або що на стіні було щось, що не відображало ІЧ-світло.

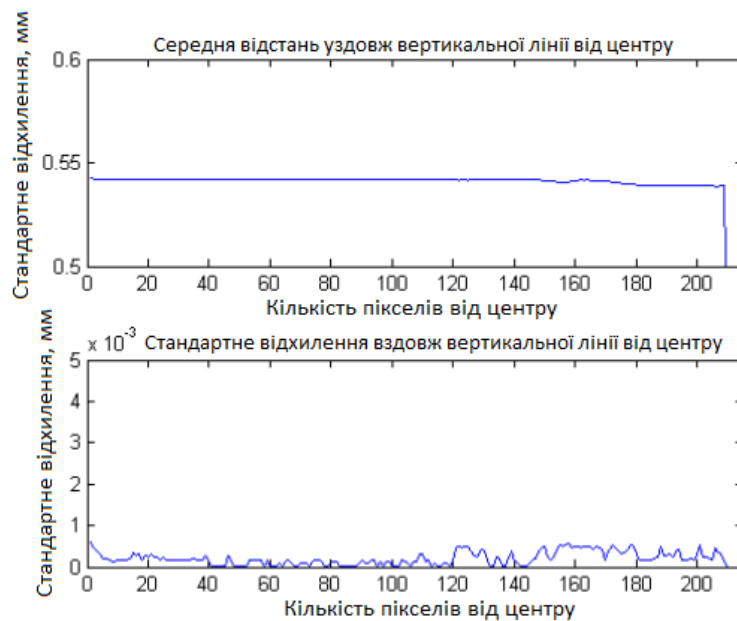
На рисунку 3.5с на облицювальній сторінці ми можемо побачити середню вимірювану відстань до стіни у тих самих 100 кадрах, які використовуються для створення рисунка 3.5а, значення в барі праворуч дано в метрах. Оскільки стіна плоска, то відстань до неї повинна бути однаковою по всьому зображенню, але ми бачимо, що в цьому результаті верхня частина зображення вимірюється трохи ближчою, ніж нижня частина зображення, а відстані вимірювані зліва направо, здається постійним. Однією з причин для різних значень згори донизу може бути те, що Kinect не було повністю перпендикулярно до стіни. Вбудований двигун Kinect був використаний для нахилу камери, щоб дивитись прямо на стіну, але може бути деяка неточність в двигуні або акселерометрі, що дає невеликий зміщення кута камери. Існує також деяка слабкість в суглобах, що тримають датчик, який може витіснити камеру.

На рис. 3.6 на наступній сторінці показано середню глибину та стандартне відхилення уздовж горизонтальної та вертикальної лінії від центру зображення та до верхньої та правої краю. Ми бачимо, що середня відстань досягає дуже стабільного від центру і до країв. Найцікавіша частина цих графіків є стандартним відхиленням, це може дати ознаку варіації вимірювань.

Особливо у вертикальній лінії стандартне відхилення дуже низьке, <1 мм. Уздовж горизонтальної лінії там, здається, невелике збільшення відхилення як відстань до центру збільшується, але для обох ліній можна сказати, що свідчення відносно стабільні і що на всю глибину зображення з Kinect, здається, дає хороші свідчення глибини. Можливо, за винятком кутів, де глибина трохи ближче тягнеться.



(а) середня відстань, виміряна пікселями уздовж горизонтальної лінії від центру пікселя та до правого краю



(б) середня відстань, виміряна пікселями вздовж вертикальної лінії від центр піксель і зверху

Рисунок 3.6 – Середні відстані уздовж горизонтальних та вертикальних ліній від центру зображення глибини

Рисунок 3.7 показує стандартне відхилення середньої вимірюваної відстані до квадрата 20x20 см на стіні. Відхилення невелике, менше 2 мм, до 1 м. Через 1 м відбувається різке збільшення відхилення. Отже, з цього видається, що якщо ви хочете отримати точні результати вимірювань, вам доведеться працювати в діапазоні 0,5 м - 1 м.

Цей результат трохи краще, але, здається, добре узгоджується з точністю, представленою в [27].

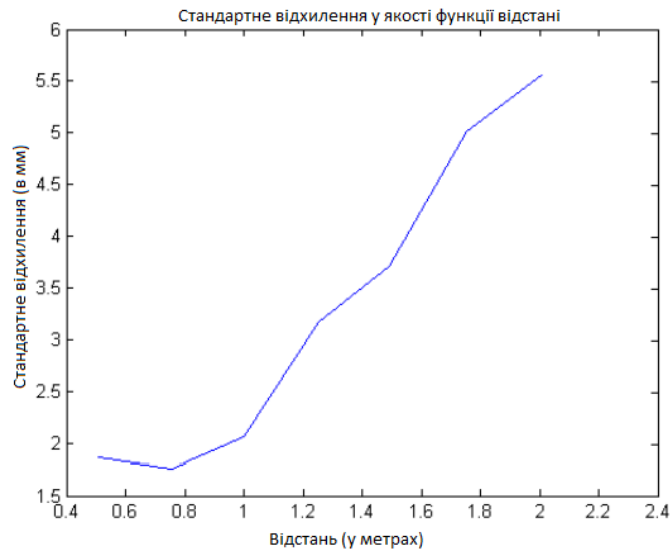


Рисунок 3.7 – Стандартне відхилення на відстані від Kinect до плоского квадрата площею 20x20 см

3.4 Точність глибини

Рисунок 3.8 показує середнє вимірювання відстаней між поверхнями, стандартним відхиленням і середньої похибкою для 100 кадрів для 7 різних відстаней між Kinect і моделлю. Ми бачимо, що в вимірюваній різниці глибин 0,4 мм - 2 мм є помилка, яка становить похибку від 1% до 4%. Випробування було виконано на декількох відстанях, щоб сказати напевно, але, схоже, настільки ж ймовірно, що виміряна відстань занадто довга або занадто коротка. Також зрозуміло, що загальна похибка збільшується з відстанню.

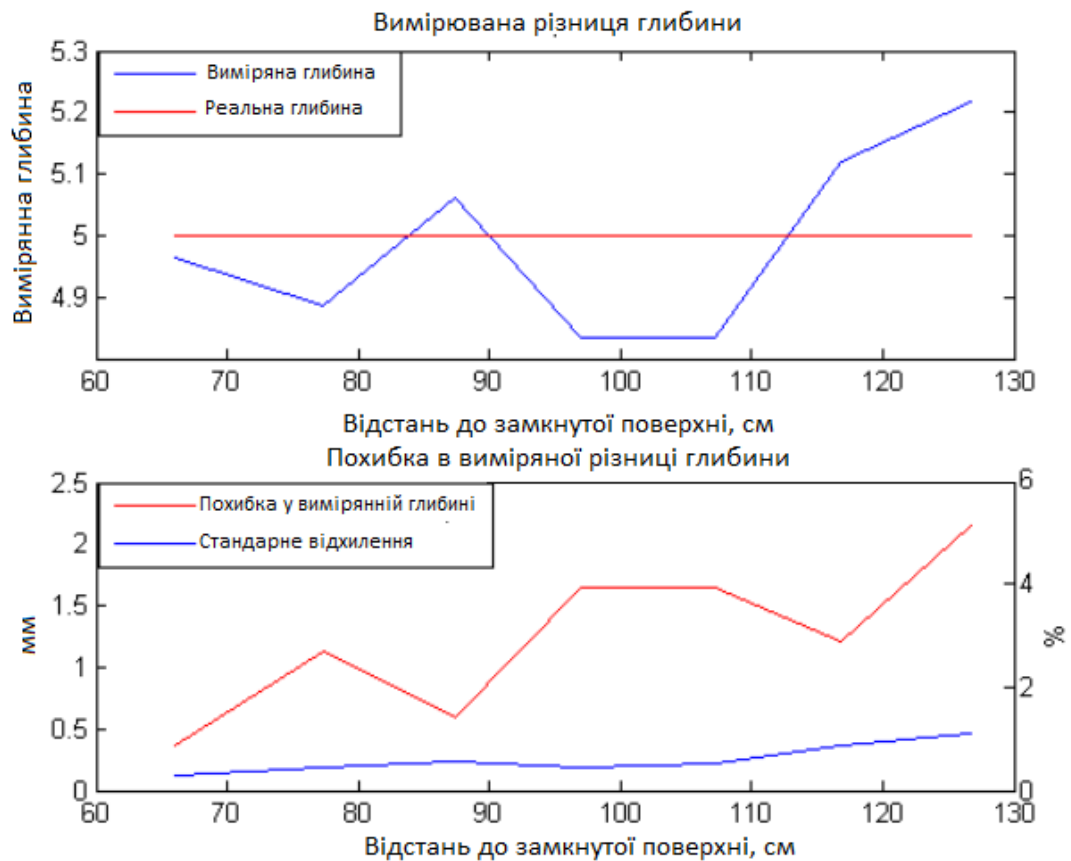


Рисунок 3.8 – Середня вимірювана різниця глибини

3.5 Точність ХУ

Як і очікувалось, точність зменшилася, оскільки збільшилася відстань між папером та датчиком. Але, оскільки дистанція збільшилася, точність зменшилася. Це суперечить очікуванню того, що точність буде приблизно однаковою, тоді як точність зменшиться з збільшенням відстані. Точність зменшена з похибки близько 5мм на найближчому вимірі та до 50мм на відстані 4м. На відстані 1 м, похибка $\approx 5\%$, і вже при 1.5 м похибка $\approx 10\%$. Слід зазначити, що цей тест заснований лише на вимірах у точках на прямих від центральної точки і дуже чутливий до шуму на краях.

3.6 Точність на краю

Графіки на рисунку 3.10 показують середню відстань та стандартне відхилення від кожного ряду пікселів на рисунках 3.3b та 3.3c. Тут ми можемо побачити, як стандартне відхилення збільшується в місці різкого краю, тоді як він стабільніший уздовж нахиленого краю. Зверніть увагу, що під час запуску з найкоротшою відстані між Kinect і моделлю, нахилений край повинен був бути близьким до Kinect, тому дані пошкоджені великою кількістю відсутніх значень глибини. З цих результатів ми бачимо, що на гострих краях слід очікувати, що для виявлення краю потрібно 2-3 рядки пікселів, і ви отримаєте піки відхилення в краях розташування. Для нахиленої поверхні немає великого піку відхилення на краях, але загалом трохи більше відхилень вздовж нахиленої поверхні, ніж на рівній поверхні. З цього можна констатувати, що точність Kinect випадає навколо краю, і що більше шуму на нахилених поверхнях порівняно з плоскими поверхнями.

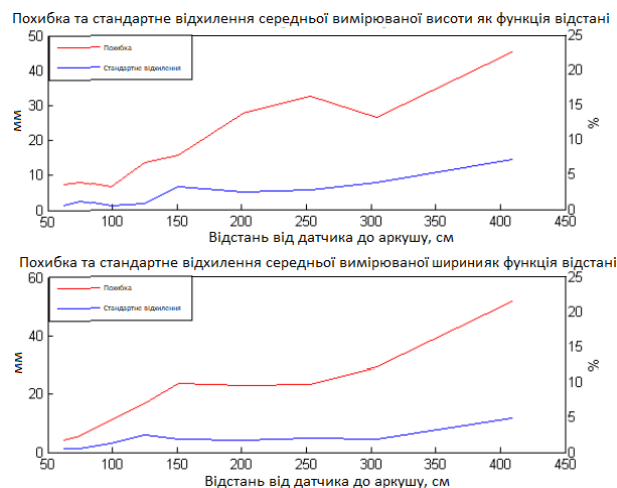


Рисунок 3.9 – Помилка вимірюваної ширини та висоти паперу формату А4, прикріпленого до вікна

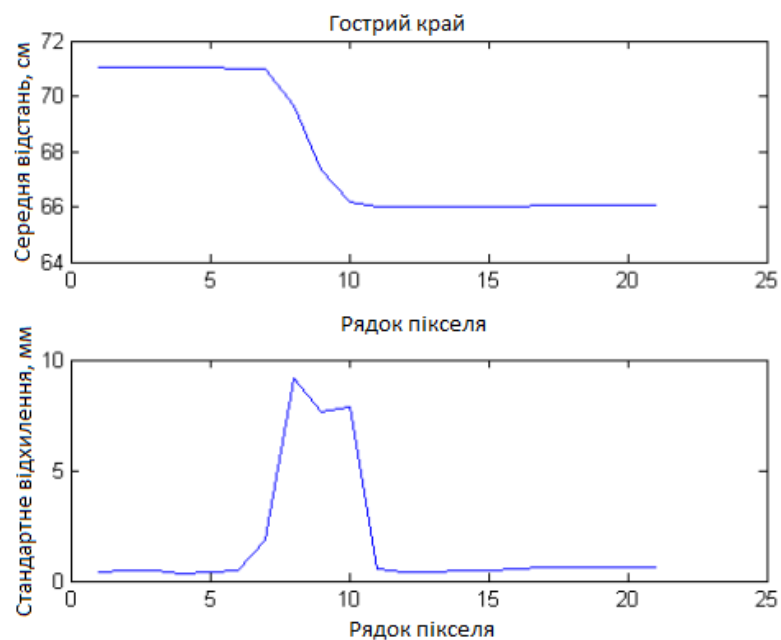
3.7 Реконструкція моделі

Розрахований радіус сфери на різних відстанях можна побачити на рисунку 3.11. Червона лінія вказує на справжній радіус сфери, тоді як синя лінія - це розрахунковий радіус. Для кожної відстані радіус розраховувався на сто кадрів, а середній радіус, який був знайдений, повідомляється. Як ми бачимо із стандартного відхилення, перше

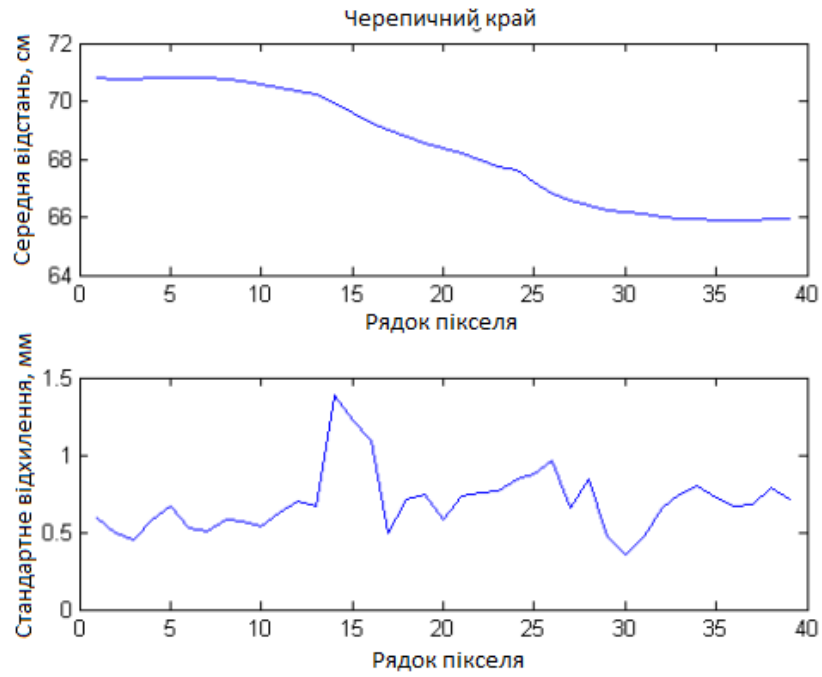
вимірювання на 55см дуже нестійке, а вимірювання від 65см до 125см стабільні. Це пояснюється тим, що перше вимірювання було на нижній межі операційного діапазону Kinect, і, отже, в читаннях було багато шуму, тоді як на інших вимірах показання були стабільними і чудовими. Найкращий результат - 65 см, де розрахунковий радіус становить 4,93 см, що лише на відстані 0,7 мм від справжнього радіусу.

Рисунок 3.12 показує результати при оцінці радіусу з увігнутої сторони сфери. Результати тут набагато кращі, ніж у попередньому тесті з випуклою стороною сфери. Але, як було зазначено вище, тести виконуються з різною установкою, тому важко дати пряме порівняння результатів.

Використовуючи таку ж установку для повної сфери, як для напівсфери, ми отримуємо результати, показані на рисунку 3.13. Тут побудовано оцінку радіусу від опуклої сторони поряд з ділянкою з рисунка 3.12. Спочатку можна помітити, що для всієї сфери є менше вимірювань, ніж для половини сфери. Це тому, що з внутрішньою сферою можна було вимірювати ближче до Kinect, ніж зовні. Існує також набагато менш відхилення в вимірах, проведених всередині сфери. Нарешті, ми зауважимо, що радіус, розрахований на дані зсередини, виглядає більш правильним, ніж зовнішні розрахунки. Отже, з цього може здатися, що Kinect дає кращі результати на увігнутих поверхнях, то на опуклих поверхнях. Можна також ближче вимірювати вигнуті поверхні, порівняно з опуклими поверхнями.



(a) Різкий край



(б) нахилена поверхня

Рисунок 3.10 – Середня глибина і стандартне відхилення в рядках пікселів на рисунках 2.3б та 2.3в

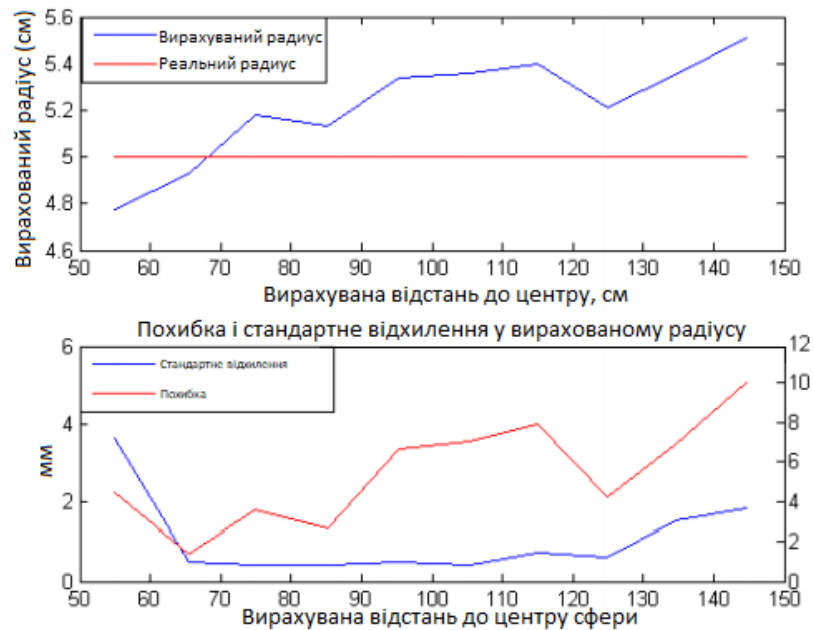


Рисунок 3.11 – Обчислений радіус від опуклої сторони сфери

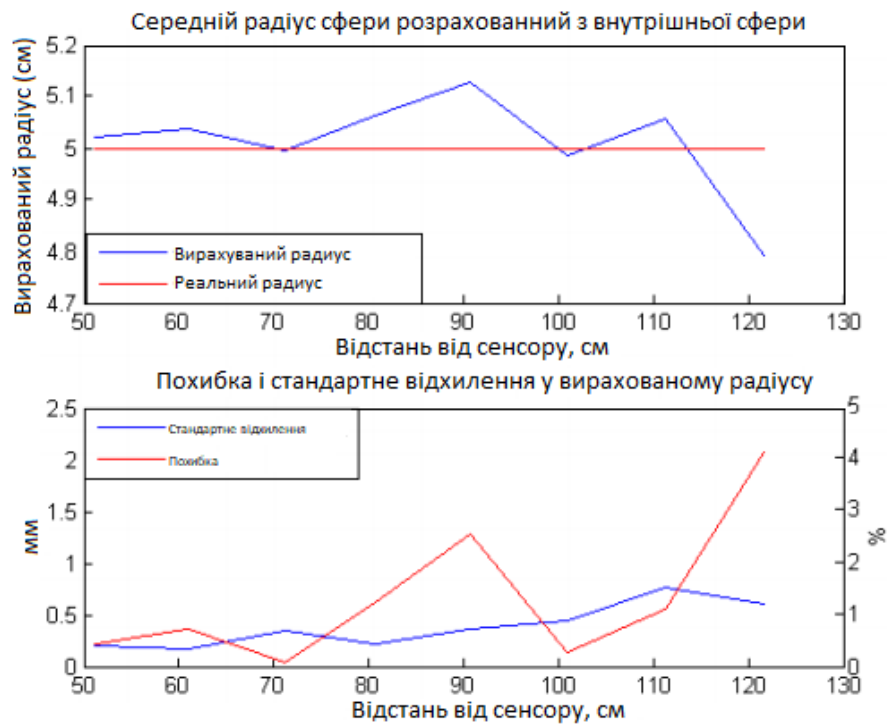


Рисунок 3.12 – Розрахунковий радіус з увігнутої сторони сфери

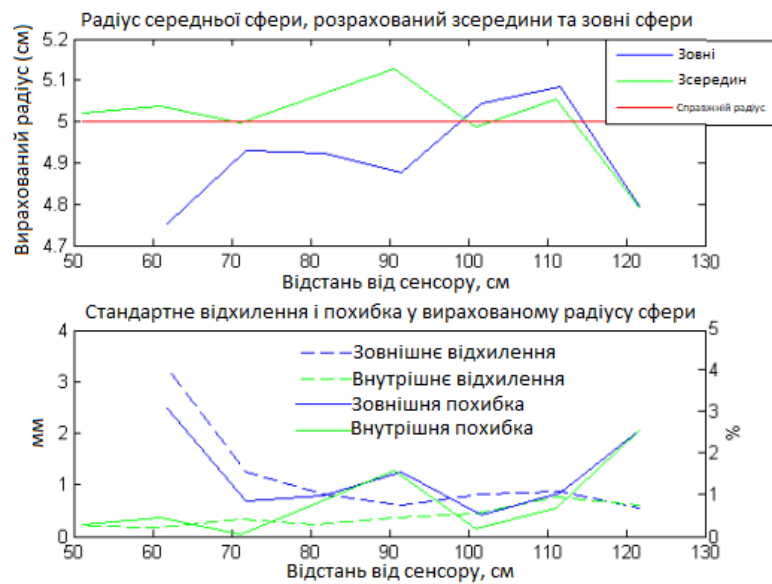


Рисунок 3.13 – Порівняння обчислених радіусів з даними всередині та за межами сфери

3.8 Кут огляду

На рисунку 3.14 ми можемо побачити кількість точок даних, які Kinect змогла сфотографувати від сфери (синьою кривою). На підставі кривої на рисунку 2.2с ми можемо побачити теоретичний максимум (червона лінія) кількості пікселів, які Kinect має

використовувати на кулі з радіусом 5 см. Світло-блакитна лінія на рисунку 5.14 на наступній сторінці є кривою сфери, яка найкраще підходить для кількості пікселів, які дійсно використовувався Kinect. Ця сфера має радіус 4,4 см, що свідчить про те, що Kinect бачив лише 88% сфери. Оцінений радіус 4,4 см можна потім помістити в рівняння (2.2). Це дає нам $\alpha = 61,64^\circ$. Отже, з цього можна припустити, що якщо α стане більшим, ніж Kinect, Kinect не зможе прочитати будь-яку інформацію про глибину з об'єкта. Але, як ми бачили на рисунку 3.3, навіть з плоскої поверхні ми отримали менше точок даних, ніж очікувалося від теоретичного дозволу. Отже, насправді Kinect, ймовірно, бачить більше сфери, ніж оцінка 4,4 см, що дає α трохи більше значення, ніж тут знайдено.

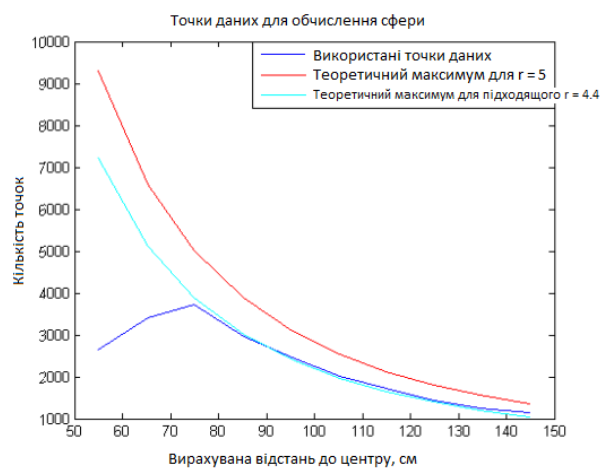


Рисунок 3.14 – Оцінка того, скільки сфери Kinect можна побачити, виходячи з кількості точок даних, взятих з Kinect.

3.9 Частотний відгук і пропускна здатність Kinect

На рисунку 3.15 на наступній сторінці показана позиція, виміряна Kinect, у порівнянні з реальною позицією інструмента робота. Обчислення реальної затримки між реальною та вимірювальною позицією є важким, оскільки дані Kinect нестабільні через погану роздільну здатність. Це особливо проблема на низьких частотах, коли робот рухається повільно і на найнижчих амплітудах. Як видно на рисунку 3.15а, виміряна відстань стрибає вгору і вниз, коли положення інструмента рухається між двома кроками квантування глибини. Через це часто не існує ніяких піків або нульових перехресть, які можуть бути використані для вимірювання затримки між фактичною позицією та положенням, повідомленими Kinect. Затримка визначається як час, який він приймає від

робота, перетинає лінію 0,45 і до тих пір, поки глибина, виміряна від Kinect, перетинає ту ж лінію. У ситуаціях, коли лінія 0,45 перетинається декількома даними даних Kinect, затримка обчислюється як середній час між сигналом, який спочатку досяг лінії 0,45, і останній раз це робиться. Коефіцієнт посилення розраховується як амплітуда пік-пік сигналу Kinect у порівнянні з пік-пік амплітуди позиції робота. На рисунку 3.15а ми бачимо, чому використовується амплітуда пік-пік замість амплітуди піку. У першому періоді на цьому графі ми бачимо, що найвищі точки вимірюваної позиції значно перевищують реальну позицію, тоді як найменша вимірювана позиція приблизно така ж, як реальна. Це означає, що прибуток буде іншим, якщо він буде розраховуватися з найвищих піків у порівнянні з найнижчими піками. Під час випробувань спостерігається тенденція до того, що верхні вершини отримують більший приріст, ніж нижчі вершини. Але є також результати, коли як високі, так і низькі вершини перевищують піки реальних рухів. Це видно на рисунку Е.1 в додаванні Е. Те, що високі піки часто забезпечують більший приріст, ніж нижчі піки, може означати, що Kinect в основному вимірює значення а на довгій відстані, ніж а до короткого. Можливо також, що вирівнювання між вимірами Kinect і роботом не є достатньо точним, але коли Kinect вимірює положення робота, перш ніж почати рух, середня похибка вимірювання була $<0,1$ мм, тому цього недостатньо для того, щоб викликати такий високий прибуток.

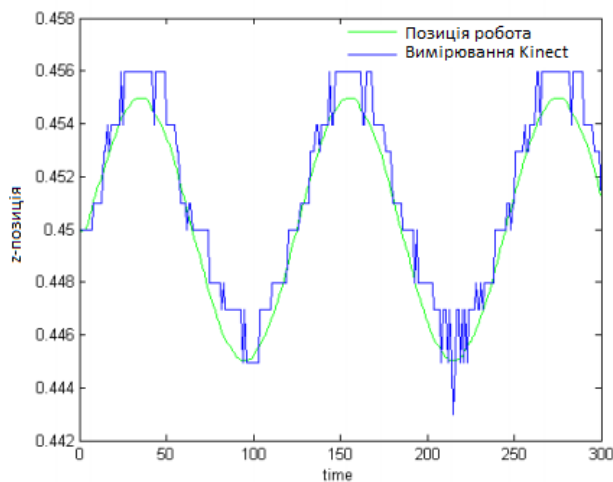
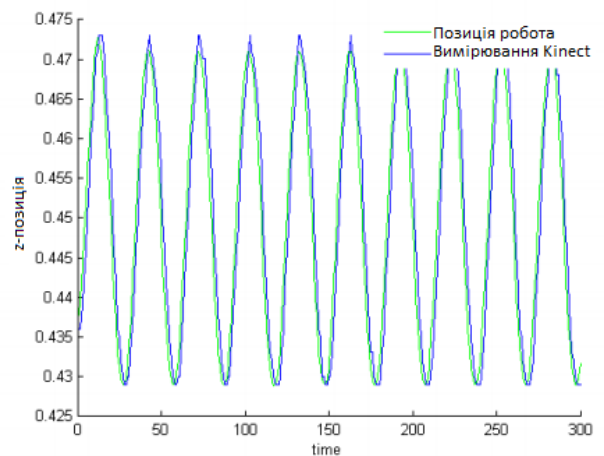
(a) $f = 0.25\text{Hz}$, $A = 3\text{cm}$ (b) $f = 1\text{Hz}$, $A = 3\text{cm}$

Рисунок 3.15 – Відстеження Kinect інструмента робота.

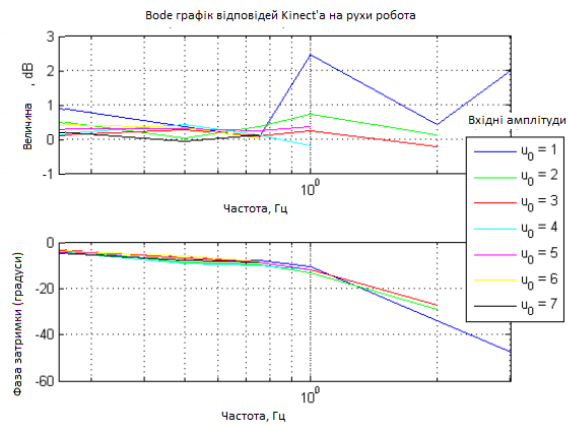


Рисунок 3.16 – Графік передбачення реакції Kinect на рух робота. " U_0 " - це пік-пік амплітуди робота було сказано, щоб слідувати.

Частотний відгук від відстеження позиції робота Kinect можна побачити на ділянці передбачення на рисунку 5.16. Зверніть увагу, що амплітуда u_0 , задана в графіку, є амплітудою пік-пік керуючого сигналу робота. Амплітуда руху робота затухає по відношенню до вхідного сигналу, тому амплітуда амплітуди руху трохи нижче, ніж амплітуди, наведені на графіку.

З рисунка 2.6 очікувалося, що коефіцієнт посилення збільшиться, оскільки збільшується частота. Це явно не так. Ідеальна відповідь була б коефіцієнтом посилення 0 дБ, але в більшості випадків коефіцієнт посилення становить > 0 дБ. Причиною цього є роздільна здатність Kinect, яка дає неточні вимірювання. Через це і так як максимальна частота тестування була обмежена до 3 Гц, то не можна сказати нічого про реальну пропускну здатність Kinect, але, схоже, Kinect може відслідковувати більш швидкі частоти, ніж робот. Це ґрунтується на тому, що Kinect не мав проблем із відстеженням робота на його максимальній частоті. Фазова затримка відносно низька від 0,25 Гц до 1 Гц, але від 1 Гц до 3 Гц вона швидко зростає. З часом середня затримка між положенням робота та позицією, виміряною Kinect, становить 0,0399 с. Це дуже добре, якщо врахувати, що частота кадрів Kinect в 30 Гц буде означати затримку 0,033 секунди. Якщо ми також знімемо потенційну затримку $1/125$ с з моменту оновлення позиції робота, ми отримаємо загальну затримку $0.0399s - 1 / 30s - 1 / 125s = -0.0014s$. Що явно неможливо, оскільки Kinect не може бути випереджаючим. Але це лише вказує на те, що затримка між оновленням робота та визначенням Kinect не була повною мірою $1/125$ с. Інша причина, чому Kinect, здається, знає позицію робота, перш ніж він з'явиться, - це погана роздільна здатність Kinect у порівнянні з роботом. Через роздільну здатність Kinect на цій відстані він може повідомити, що робот ближче або далі, ніж це дійсно є, і це схоже на те, що Kinect випереджає робота.

Нарешті, слід зазначити, що всі дані від Kinect засновані на показаннях з одного пікселя і, таким чином, більш привабливими для шуму, ніж якщо глибина була виміряна з зваженої суми декількох пікселів. У розділі 3.1 також було показано, що збільшення глибини Kinect збільшується, якщо вимірювання приймаються з кількох пікселів. Відповідь може бути кращим, якби це було зроблено.



(а) нерухомий об'єкт на близькій відстані

(б) рухомий об'єкт на близькій відстані

Рисунок 3.19 – Різниця в дійсних вимірах глибини від рухомого та нерухомого об'єкта на тій же відстані

3.10 Інші спостереження

Коли об'єкти наближаються до Kinect, це є показники глибини з точки зору Kinect, який спочатку пошкоджений. Також було помічено, що ця корупція є більшою проблемою на об'єктах, що стоять на місці, ніж на об'єктах, що мають малий безперервний рух. Це видно на рисунку 3.19, в 3.19а об'єкт стоїть у нижній межі операційного діапазону Kinect, тоді як в 3.19b об'єкт має таку ж відстань до Kinect, але постійно переміщається ліворуч і праворуч. Ми бачимо, що різниця в дійсних вимірах глибини масивна. Це може свідчити про те, що Kinect працює краще на рухомих об'єктах, ніж на нерухомих об'єктах.

4 ІНТЕРФЕЙС KINECT-РОБОТУ ТА РЕЗУЛЬТАТИ ТЕСТУ

4.1 Робот

Робот має інтерфейс у режимі реального часу, який може бути використаний для підключення робота до комп'ютера та контролю за ним. Цей інтерфейс з комп'ютером дає робототехніці дві контрольні шари. По-перше, на комп'ютері Unix є контролер, який надсилає роботам швидкісні команди в реальному часі для кожного з'єднання. Тоді робот має внутрішній контролер, який посилає струм до двигуна в кожному шарнірі, на основі команди швидкості. Контролер робота надсилає дані на комп'ютер Unix, ці дані описують поточну позицію суглобів, швидкість тощо. Це зв'язок між комп'ютером і роботом працює на частоті 125 Гц. На малюнку 4.1 показано проста ілюстрація того, як система налаштована. Kinect надсилатиме інформацію про 3D-зображення на комп'ютер, де позиційні команди отримуються з зображеннями і надсилаються роботодавцю. Дані Kinect обробляються у власній обробці потоку, яка працює на частоті 30 Гц.

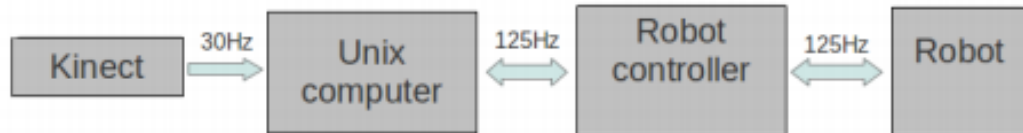


Рисунок 4.1 – Модель робототехнічної системи з Kinect

При використанні команд позиції, орієнтація інструмента робота відносно робочої бази повинна бути визначена оператором. З позиції команди, зворотна кінематика робота використовується для розрахунку кута, необхідного для кожного з'єднання, щоб розмістити інструмент у правильному положенні. Для цього робота існує вісім можливих різноманітних кутових конфігурацій, які додають робота у правильне положення. У ситуаціях, коли є декілька рішень, вибирається рішення, найближче до поточного обертового кута з'єднання. Система використовує пропорційний контролер для розрахунку команд швидкості для кожного з'єднання.

Швидкість робота контролюється пропорційним контролером [41, с.216]. Таким чином, швидкість визначається різницею в потрібній позиції, а поточна позиція помножена на попередньо визначений коефіцієнт посилення (4.1). P_d та P_c - це бажана і поточна позиція інструмента робота.

$$\text{Out} = (\text{Pd} - \text{Pc}) * \text{Gain}. \quad (4.1)$$

Це призводить до швидкого руху робота, коли помилка між поточною та бажаною позицією велика, і повільніше, коли робот закривається в цільовому положенні. Великий приріст забезпечує швидкий рух та кращу реакцію в системі, але якщо прибуток до великих, вихід може бути нестійким. Отже, потрібно встановити коефіцієнт корисної дії, який дає найкращу відповідь на вхідний сигнал і як і раніше виробляє стабільний вихід. Варто також відзначити, що при пропорційному контролері завжди буде помилка між бажаною позицією та поточною позицією [41, с.136], якщо для коефіцієнта посилення не встановлено значення ∞ , що, очевидно, неможливо.

4.2 Kinect - інтерфейс робота

Існуючий інтерфейс між комп'ютером і роботом є основою при створенні інтеграції між роботом і Kinect. Більшість інтерфейсів комп'ютера-робота написано на C ++, тому додатковий інтерфейс Kinect записується як простий файл заголовка C ++ на базі OpenNI і PCL. Вимоги до інтерфейсу до роботи - це робоча установка OpenNI і PCL. Встановлюючи попередньо встановлені бінарні файли з веб-сайту PCL, всі інші залежності автоматично включаються та встановлюються. Ідея за інтерфейсом полягає в тому, що нова команда позиції робота буде виводитися з карти глибини, зафіксованої Kinect. Інтерфейс пропонує методи для запуску та зупинки Kinect, переведення точок з координатної системи Kinect в систему координат робота та отримання останньої позиції, що обчислюється Kinect. Алгоритм керування, який знаходить наступну позицію робота, повинен перейти до того, щоб користувач здійснив, щоб він / вона отримав саме потрібну функцію. Код інтерфейсу можна побачити в додатку В.

Kinect використовує кілька секунд, щоб розпочати і калібрувати себе, щоб дати точні вимірювання глибини, тому рекомендується, щоб інтерфейс Kinect ініціалізувався одночасно з роботою системи. Щоб запустити Kinect, слід використовувати функцію `initializeKinect ()`. Ця функція встановлює глобальну змінну для попередньо визначених значень і відкриває з'єднання з Kinect за допомогою `openNIGrabber PCL`. Інтерфейс PCL вимагає функції зворотного виклику, до якої можна доставити нові дані з Kinect. Ця функція зворотного виклику - це хмара `cb (const pcl :: PointCloud <PointT> :: ConstPtr & cloud)`. PointT говорить, який тип точкового хмарного PCL повинен бути отриманий від

Kinect, в цьому випадку PointT сидить на PointXYZ, який дає дані тільки з датчика глибини. PointT також може бути розміщений на PointXYZRGBA, який дасть дані глибини, вирівняні за даними RGB з камери RGB на Kinect. Коли встановлено інтерфейс Kinect, і функція зворотного виклику підключена до процесу Kinect, Kinect запускається з викликом *kinectInterface* \rightarrow *start* (). Почнеться новий процес, який отримує дані від Kinect за 30 кадрів на секунду. Коли новий кадр готовий, він повинен бути відправлений до функції *findNextPosition* (), де користувач повинен ввести алгоритм керування. Щоб зберегти потужність обробки, цей алгоритм управління повинен знаходити наступну позицію в системі координат Kinect, а потім викликати функцію *transformPoint* () для перетворення цієї єдиної точки з координатної системи Kinect в базову систему координат робота. Це заощадує деякий час обробки, оскільки тільки одна точка повинна бути перетворена, а не всі потенційні дані 307200, які може містити один кадр. Нова позиція зберігається методом *transformPoint* (), а потім робот-контролер може використовувати *kinectGetLastPosition* (), щоб отримати останню позицію від Kinect. Змінні, що зберігають наступну позицію, поділяються процесом Kinect і роботом контролера, тому для того, щоб обидва процеси мали доступ до цієї змінної, одночасно використовується об'єкт *boost :: mutex1*. Щоб зупинити Kinect і закрити інтерфейс, слід використовувати функцію *stopKinect* (). Якщо процес керування роботом повинен намагатися прочитати нову команду позиції одночасно з тим, як процес Kinect зберігає його, робот отримає останню знайдену позицію.

Матриця перетворення, *kinectTransMat*, використовувана для перетворення точок даних з координатної системи Kinect на базу робота, повинна бути визначена користувачем. На даний момент це потрібно зробити в вихідному коді в функції *initializeKinect* (). З поточними конфігураціями ця матриця визначається як одинична матриця, тому обертання не буде виконуватися. Процес калібрування, щоб знайти ідеальну матрицю перетворення між Kinect і роботом ще не розроблено. Масив *kinectRotation* зберігає обертання, інструмент робота повинен мати відносний бази робота. У реалізаціях, що використовувались для керування роботом з Kinect, це поворот було збережено. При поточній установці інструмент z-осі буде постійно вказувати на протилежний шлях робочої бази z-осі. Це повинен бути реалізований код, який також встановить обертання інструменту в потрібну позицію, що дозволяє змінювати обертання при зміні середовища.

4.3 Kinect - пропускна здатність робота системи

Принцип частотної характеристики та пропускної здатності пояснено в 4.8. Наступне випробування має на меті знайти пропускну здатність системи Kinect-robot. Замість того, щоб підключити Kinect до робота, процес, який імітує Kinect, був доданий до контролера робота на комп'ютері Unix. Процес генерує синусоїдальну хвилю і використовує це для надання позиції програм робота, програма змінить лише z-позицію інструмента робота. Під час тесту положення x- і y не змінюватимуться. Процес буде працювати на частоті 30 Гц, що буде таким самим, як Kinect. Команди позиції $z(t)$ будуть оновлюватися, беручи зразки з синуса на частоті 30 Гц за допомогою рівняння (4.2), де A та f - амплітуда і частота синуса, а t - час, який вважається тактовим циклом у модельованому процесі Kinect.

$$z(t) = A * \sin(t * 2\pi * f) / 30. \quad (4.2)$$

Подібно до випробувань смуги пропускання Kinect, частота поступово збільшувалася, і для кожної частоти тест проводився для збільшення амплітуд. Для кожної комбінації частоти та амплітуди випробування пройшло 300 тактових циклів у процесі Kinect, що дорівнює 10 секунд. Тест починається з інструмента робота в заздалегідь визначеному положенні за замовчуванням (-0.5, -0.11, 0.45), тому B додається до положення позиції, що дозволяє інструменту рухатися вгору та вниз навколо положення.

```

for f = fmin → fmax do
for A = Amin → Amax do
for t = 0 → 300 do
nextRobotPos ← B + A*sin(t*2π* f) / 30
end for
end for
end for

```

Програма 4.1 – Алгоритм з перевірки пропускної здатності робота

Ця позиція за замовчуванням також вибирається, щоб гарантувати, що робот має вільне місце для переміщення та уникнення зіткнень. Для додаткової безпеки навколо інструмента робота визначено обмежувальну коробку розміром 20x20x20 см, з центром у позиції за замовчуванням інструмента. Якщо інструмент робота досягає будь-якого

обмеження в обмежувальній коробці, його буде зупинено. Кожна координата осі z, яка змінюється під час тесту, координата x і y залишається незмінною, тому інструмент робота повинен рухатися прямо вгору та вниз для всього випробування. Для цих тестів коефіцієнт підсилення в пропорційного контролера був встановлений на 9. Це значення було вибрано з візуального огляду руху робота. Коли коефіцієнт посилення був встановлений вище 9, рух роботу, коли він закривався в цільовому положенні, коливався. На додаток до коефіцієнта посилення, необхідно встановити команду максимальної швидкості, яку комп'ютер може відправити роботу. Максимальна швидкість, з якою робот може рухатися перед зупинкою, з причин безпеки - π радіан на секунду. Таким чином, щоб уникнути запобігання безпеці, максимальна швидкість була обмежена до 3 радіана на секунду.

4.4 Демонстрація того, як Kinect безпосередньо контролює положення робота

Цей остаточний тест був демонстрацією того, як позиціонування робота в інструменті може бути безпосередньо кероване Kinect. Демонстрація призначена для демонстрації того, як Kinect може використовуватися для відстеження поверхні та створення робочих точок, щоб вони могли стежити за поверхнею. Kinect розташовано нижче і трохи на бік інструмента робота. Потім картонна плита, спочатку тримана на тій же висоті, що й інструмент робота, повільно просувається вгору і вниз над Kinect. Приймаючи вимірювання глибини від центрального пікселя зображення глибини, отриманого від Kinect, командні позиції надсилаються до робота. Як і в ході тестування пропускної здатності Kinect, дані з Kinect та початкового положення робота вирівнюються, дозволяючи Kinect спочатку виміряти відстань до пластини, коли вона знаходиться поруч із інструментом.

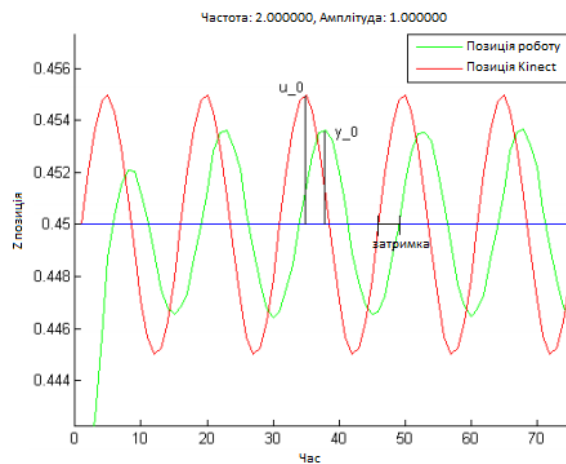
Ідеальним для цього випробування буде мати ще одного робота. Потім один робот міг рухатися, а другий міг йти за цим рухом за допомогою Kinect. Потім час і фактичне положення руху можна було б зберегти разом з вимірами від Kinect і положення робота, і точне відстань може бути розраховане з цього.

4.5 Kinect - пропускна здатність системи роботів

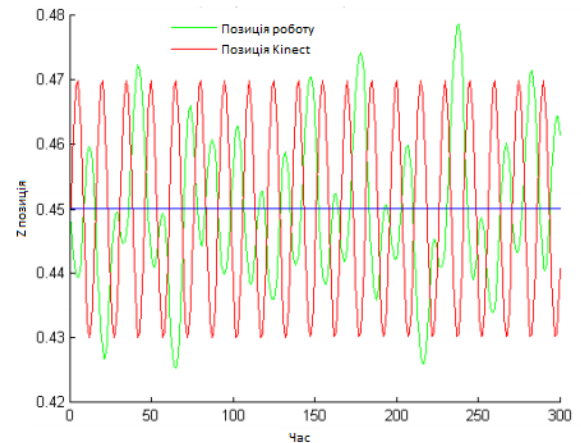
На рис. 4.2а показано, як робот реагує на вхід з $f = 2\text{Hz}$ і $A = 1\text{см}$. Ось x на графіках являє собою час, заданий як тактові цикли в процесі Kinect. На цьому графі ми можемо побачити, як позиція робота добре слідує за командам позиції, але існує деяка затримка і реакція затухає в порівнянні з вхідним сигналом. На малюнку 4.2б показано, як робот рухається, коли частота та амплітуда досягають високих значень, для чого слідує. Рух робота ніколи не стабілізується, і ясно, що він не може працювати при такому поєднанні частоти та амплітуди. Таблиця 4.1 показує результати всіх варіантів f та A , перевірених для. X означає, що вихід ніколи не стабілізувався, як показано на малюнку 4.2b, поки – вказує, що для цієї комбінації f та A не було виконано тестування. $|H(j\omega)|$ розраховується з середнього коефіцієнта підсилення між усіма парами верхніх піків, а $\angle H(j\omega)$ - середня затримка між кожним часом, коли контрольна позиція перетинає лінію 0,45, а позиція робота перетинає ту ж лінію. Всі значення розраховуються після того, як рух робота стабілізувався. Для $f = 3/4$ та $A = 10$ см робот дійшов до нижньої межі коробки, що обмежує безпеку, і тест було зупинено. Це означає, що коли робот зупинився, інструмент був у залі нижче, ніж 0,35, а з максимальною амплітудою вершина-до-вершини 10 см, то зрозуміло, що робот перемістився на 5 см вище за призначення. Це, мабуть, пов'язано з тим, що коефіцієнт підсилення в пропорційного контролера встановлений високо, завдяки чому робот рухається далеко і швидко, і таким чином рухається мимо бажаного положення, перш ніж мати шанс зупинитися.

У всіх тестах, з різними частотами та амплітудами, часова затримка в системі була дуже стабільною, в основному залишаючись між 2/30 і 3/30 с, максимальна затримка на низьких частотах і загальна середня для всіх циклів 0,0952с. Цілком природно, що часова затримка трохи менше на більш високих частотах, оскільки помилка між бажаною позицією та фактичною позицією робота зростає з більшою частотою. З вищою помилкою, вихід з пропорційного контролера збільшується, повідомляючи роботу швидше рухатися. Але навіть якщо затримка в часі менша, фазова затримка зростає з більшою частотою. Це видно з графіку Боде з результатів на рисунку 4.3. У ділянці Боде ми можемо чітко бачити, що твердження, зроблене [41], є істинним. Коли вхідна частота зростає, коефіцієнт підсилення на виході зменшується. Для кожної частоти величина відповіді є в основному стабільною, незалежно від заданої амплітуди, але, оскільки частота збільшується, виникає менше амплітуд, якими система може справлятися. Також фазова затримка, схоже, збільшується з амплітудою. Якщо ми визначимо частоту відсіку

системи, де величина перетинає лінію -3 дБ, то ми можемо побачити, що це відбувається при ≈ 2 Гц. Але на частоті 2 Гц максимальна амплітуда, на якій він керується, становить 2 см, тому це не дуже хороший результат. Оскільки інструмент робота має зареєстровану максимальну швидкість ≈ 1 м / с, очікувана відповідь була кращою. Звичайно, не можна очікувати максимальної швидкості з усіма змінами напрямку, коли робот управляється синусоїдальною хвилею. 1 м / с - це, мабуть, максимальна швидкість руху по прямій дорозі.



(а) Стабільна реакція системи



(б) Нестабільна реакція системи

Рисунок 4.2 – Стабільна та нестабільна частотна характеристика системи. Час подано у вигляді числа тактових циклів у процесі Kinect

Як згадувалося раніше, коефіцієнт корисності пропорційного контролера впливає на відповідь системи. На рис. 4.4 на наступній сторінці показано приклад того, що станеться, якщо значення коефіцієнта підсилення буде встановлено на висоту. Графік показує відповідь робота, коли сигнал керування має частоту 1 Гц, а амплітуда змінюється від 7 см до 8 см. У інтервалі часу між 0 та 300 амплітуда становить 7 см. Спочатку здається, що система збирається стабілізувати, але, коли час досягає 200, ми бачимо, що амплітуда руху робота починає збільшуватися, виходячи за межі керуючого сигналу. Коли час = 300, амплітуда керуючого сигналу збільшується до 8 см. Тепер ми можемо спостерігати, що за кожний період амплітуда робочого руху збільшується, поки вона не буде настільки далеко від сигналу управління, що захист припиниться, оскільки він порушує обмежувальну коробку.

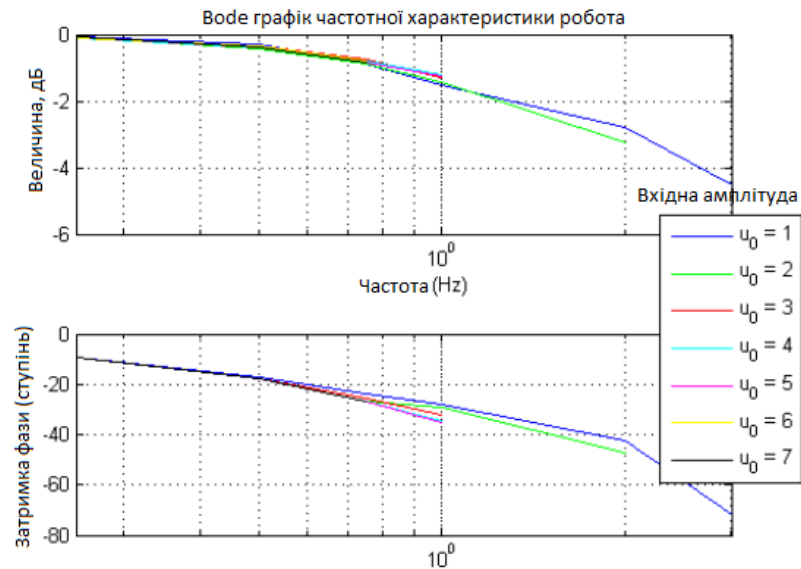


Рисунок 4.3 – Bode графік діаграми частотної характеристики, u_0 - амплітуда вершина-до-вершини (в см).

Через затримку між керуючим сигналом та рухом робота завжди буде помилка у положенні робота. Коли амплітуда збільшується, то і затримка збільшується також. А оскільки швидкість робота контролюється цією помилкою, швидкість збільшується. Таким чином, швидкість стає настільки високою, що робот рухається за межі цільової позиції, перш ніж він зупиниться своєчасно. Це ще більше збільшує похибку положення, і швидкість ніколи не стабілізується. З меншим коефіцієнтом посилення робота, ймовірно, зможе відслідковувати сигнали з високою амплітудою на низьких частотах, але це зменшить відповідь на більш високих частотах. Таким чином, зрозуміло, що коефіцієнт посилення повинен бути обраний, виходячи з бажаних амплітуд і частот, які система повинна відслідковувати.

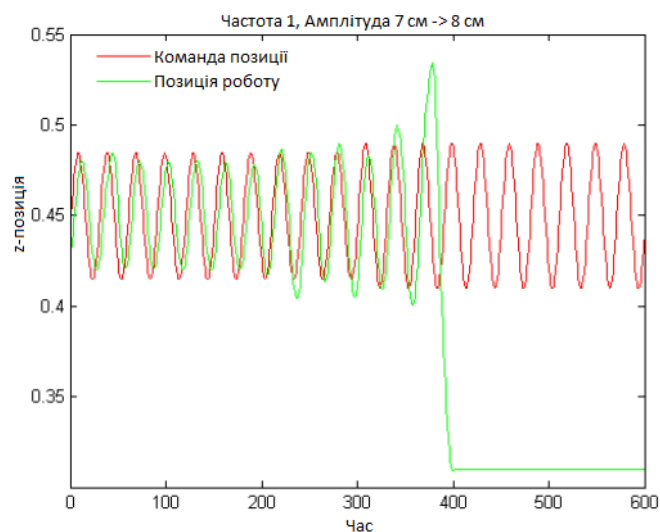


Рисунок 4.4 – Рух робота з високим коефіцієнтом посилення

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення. Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [40] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до

порушення ДНК).

Відповідно до [41], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

5.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;

– індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення мережі;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (5.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [42]: $Z_T / 3 = 0,1$ Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(5.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (5.2)$$

де $R_\Pi = R_\phi + R_0$ - сумарний активний опір фазного R_ϕ і нульового R_0 дротів, Ом;

X_Π - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5$ мм, тоді згідно таблицям 5 і 6 [42], має такий опір:

$X_\Pi = 0,11$ Ом;

$R_\phi = 7,55$ Ом;

$R_0 = 7,55 \text{ Ом}$.

Отже, $R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}$.

Тоді по формулі (5.2) знаходимо повний опір петлі:

$$Z_{\Pi} = \sqrt{15,1^2 + 0,1^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (5.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (5.4.) :

$$I_n = P / U, \quad (5.4)$$

де $P = 220 \text{ Вт}$ - споживана потужність;

$U = 220 \text{ В}$ - робоча напруга;

$k = 3 \text{ А}$ - для плавких вставок.

Отже, $I_n = 220 / 220 = 1 \text{ А}$.

Підставивши значення у вираз (5.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

5.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [50], ДБН, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [40] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 5.1).

Таблиця 5.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в ДБН (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для

оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибрати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплєну).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту ТСО'03.

ТСО'03 пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [43].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [47] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (5.5) розрахунку

штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (5.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 5.1.

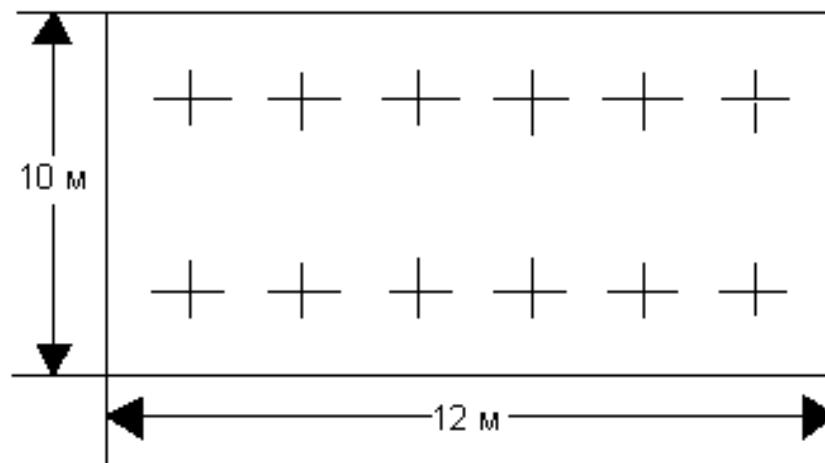


Рисунок 5.1 - Схема розташування світильників

5.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно [49] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акролеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [44] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізолюване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, ломы, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

5.5 Вплив на навколишнє середовище

В даний час зростає кількість комп'ютерної техніки в усіх галузях діяльності людини. Багато користувачів і виробників помиляються, вважаючи, що зі зменшенням і удосконаленням комп'ютерів, зменшиться їх негативний вплив на навколишнє середовище.

На даний момент найбільш суворим з існуючих світових стандартів екологічності для комп'ютерної техніки є стандарт ТСО-99. У порівнянні з попередніми він містить додаткові обмеження по частині екології, ергономіки, енергоспоживання і емісії пристроїв.

Організація по захисту навколишнього середовища Greenpeace з 2006 року оцінює виробників електроніки за кількістю важких металів і отруйних речовин, наприклад інгібіторів горіння, використовуваних ними при виробництві (інгібітор - речовина, присутність якого в невеликих кількостях призводить до запобігання або уповільнення процесів горіння або корозії; інгібітори знижують швидкість хімічних реакцій або пригнічують їх). Однак навіть оцінки такої організації, як Greenpeace, не можуть претендувати на об'єктивність. Адже в одних випадках вона використовує перевірену інформацію, що стосується, наприклад, заходів щодо утилізації відходів, а в інших спирається тільки на дані виробника. А якщо компанія не повідомляє ніяких відомостей, то автоматично опиняється на нижніх рядках рейтингу. Крім того, енергетичні витрати на виробництво і перевезення продукції також необхідно враховувати при оцінці екологічної ефективності. Адже часи, коли техніка виготовлялася тільки на одному заводі, давно пройшли. Сьогодні окремі комплектуючі закупаються на різних підприємствах по всьому світу, після чого здійснюється складання пристроїв. Тому найчастіше навіть самі компанії не можуть знати, які шкідливі речовини потрапляють в атмосферу при виготовленні їх продукції і які саме метали або токсини в ній містяться.

ЖК-екрани - один з джерел парникових газів, які набагато шкідливіше діоксиду вуглецю. Рідкокристалічні монітори швидко знайшли популярність, прийшовши на зміну громіздким ЕПТ-моделям. І це не дивно, адже вони мають тонкі корпуса і споживають значно менше електроенергії. За іншим аспектам екологічної безпеки дисплеї на основі рідких кристалів також вважалися проривом, тому що в них не використовувався газ, що містить свинець. Досить довго ніхто не звертав уваги на застосовуваний для чищення РК-панелей тріфтористий азот (NF_3), і тільки в середині 2008 року вченими було доведено наявність даної хімічної речовини в атмосфері. Відкриття було вражаючим: порівняно з діоксидом вуглецю (CO_2) NF_3 має в 17 000 разів більше активного парникового газу, а його атмосферний час напіврозпаду може складати від 550 до 740 світлових років (у CO_2 - від 30 до 40 років). Закону, який обмежував би рівень викиду NF_3 , поки не існує.

Виявлення енерговитрат є таким же проблематичним процесом, як і визначення кількості матеріалів, придатних для вторинної переробки, і важких металів, що містяться в пристроях. Таким чином, надійним показником екологічності залишається тільки рівень енергоспоживання.

Полівінілхлорид, що позначається зазвичай аббревіатурою ПВХ, - це різновид пластику, що застосовується в самих різних цілях. З нього зроблена зовнішня оболонка кабелів, якими з'єднуються пристрої, він оточує електричний провід портативного комп'ютера. Це дешевий, міцний і вельми поширений матеріал. Разом з тим, за словами IT-аналітика «Грінпіс» Кейсі Харрелл, «ПВХ - найгірший з пластиків». Він є причиною виникнення гормонального дисбалансу, проблем в репродуктивній сфері та різних форм раку. Полівінілхлорид практично неможливо правильно утилізувати. Внаслідок старий матеріал виявляється зазвичай на звалищі з відходами або, того гірше, спалюється з метою вилучення мідних жил і інших цінних компонентів. При його згорянні утворюється вкрай шкідливий канцерогенний діоксин. Звалища і хімічні поховання забруднюють джерела води. Єдиний спосіб правильно утилізувати ПВХ полягає в тому, щоб відправити його в центр небезпечних відходів.

Залишається лише сподіватися, що настане час, коли технології будуть допомагати людині, не завдаючи незворотної шкоди здоров'ю навколишнього середовища.

ВИСНОВКИ

У цій роботі було проведено набір тестів для дослідження ефективності Kinect, коли мова йде про точність, точність і пропускну здатність. Kinect показав, що найбільш точні та точні на коротких відстанях, тому вимірювання повинні бути максимально наближеним. Але важливо пам'ятати, що коли дистанція зменшується, то і поле Kinect зору теж. Точність і роздільна здатність швидко зменшуються з відстанню, тому вимірювання глибини слід тримати на відстані менше 1,5 м, це дасть точність глибини > 90%. Піксель за пікселем має деякий шум, тому точність буде кращою, якщо генерувати модель гладкої поверхні замість використання прямих значень глибини від Kinect. Границя верхньої смуги пропускання Kinect не була знайдена, але було зафіксовано затримку між рухом і повідомлення про позицію, що Kinect виявився рівним $\approx 0,04$ с. При низьких частотах (<1 Гц) система Kinect-робот могла виконувати команду періодичної позиції з максимальною амплітудою до 10 см, тому не повинно бути проблем із відстеженням нормальних рухів. Фазова затримка системи була вищою, ніж очікувалося, але загальна часова затримка виявилася < 0.2s, тому затримка знаходиться в межах того, що дозволено.

Коли доходить до компенсації руху, Kinect не може бути використаний, це виходить з того, що Kinect не зможе побачити поверхню, і створення точної моделі на основі навколишньої поверхні буде важким. Тому сигнал іншого датчика буде кращим джерелом зворотного зв'язку для компенсації руху. Але Kinect все ще може бути використаний, щоб переконатися, що інші частини робота зберігають безпечну відстань від цілі.

Візуальні дані від Kinect найкраще використовувати для маневрування робота до правильної цілі, тоді як сигнали інших датчиків повинні використовуватися для остаточного позиціонування. Але це не є остаточним висновком, бо в роботі Массачусетського технологічного інституту[8] був знайдений спосіб, як покращити точність даних Kinect у 1000 разів завдяки поляризованому світлу.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці, розглянутий можливий вплив на навколишнє середовище.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) http://www.ros.org/wiki/kinect_calibration/technical.
- 2) http://en.wikipedia.org/wiki/History_of_video_games.
- 3) [www.wikipedia.org/wiki/Cathode ray tube amusement device](http://www.wikipedia.org/wiki/Cathode_ray_tube_amusement_device).
- 4) <http://classicgames.about.com/od/history/tp/ClassicVGTimelinePart1.htm>.
- 5) <http://classicgames.about.com/od/classicvideogames101/p/MagnavoxOdyssey.htm>.
- 6) [http://en.wikipedia.org/wiki/History_of_video_game_consoles_%28second generation%29](http://en.wikipedia.org/wiki/History_of_video_game_consoles_%28second_generation%29).
- 7) [http://en.wikipedia.org/wiki/History_of_video_game_consoles_%28fifth generation%29#Transition to 3D](http://en.wikipedia.org/wiki/History_of_video_game_consoles_%28fifth_generation%29#Transition_to_3D).
- 8) <http://news.mit.edu/2015/algorithms-boost-3-d-imaging-resolution-1000-times-1201>
- 9) Лоскутников А. А., Сенюшкин Н. С., Парамонов В. В. Системы автоматического управления БПЛА // Молодой ученый. - 2011. - №9. - С. 56-58.
- 10) http://www.pbs.org/tesla/II/II_robots.html.
- 11) <http://www.thocp.net/reference/robotics/robotics2.htm#26>.
- 12) <http://www.computerhistory.org/revolution/artificial-intelligencerobotics/13/289>.
- 13) <http://en.wikipedia.org/wiki/Telerobotics>.
- 14) http://en.wikipedia.org/wiki/Autonomous_robot.
- 15) <http://communities.canada.com/ottawacitizen/blogs/defencewatch/archive/2011/03/17/video-game-technology-used-to-counter-ieds.aspx>.
- 16) <http://www.physorg.com/news/2010-12-air-playstation-3supercomputer.html>.
- 17) <http://www.ifixit.com/Teardown/Microsoft-Kinect-Teardown/4066/>.
- 18) <http://www.mathworks.com/matlabcentral/fileexchange/34129.99>
- 19) Kinect - новый датчик для роботов? // <http://robocraft.ru/blog/news/>
- 20) Accuracy of the Microsoft Kinect™ for measuring gait parameters during treadmill walking Gait & Posture, Volume 42, Issue 2, 2015, Xu Xu, Raymond W. McGorry, Li-Shan Chou, Jia-hua Lin, Chien-chi Chang, pp. 145-151
- 21) J. Faust, C. Simon, and W.D. Smart. A video game-based mobile robot simulation environment. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, oct. 2006, pages 3749–3754.
- 22) L. Gallo, A.P. Placitelli, and M. Ciampi. Controller-free exploration of medical image data: Experiencing the kinect. In Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on, pages 1–6, june 2011, pages 1–6.

- 23) Guyton and Hall. Textbook of Medical Physiology. Elsevier Saunders, 11th edition, 2006.
- 24) M. Hebert. Active and passive range sensing for robotics. In Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, volume 1, 2000, pages 102 –110 vol.1.
- 25) Timo Kahlmann. Range Imaging Metrology: Investigation, Calibration and Development. PhD thesis, University of Hannover, 2007.
- 26) K. Khoshelham. Accuracy analysis of kinect depth data. In ISPRS Workshop Laser Scanning 2011, Calgary, Canada, August 29-31 2011.
- 27) Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. Int. J. Rob. Res., sep 2011, 30(11):1311–1327.
- 28) Microsoft. Kinect™for windows®sdk beta, programming guide, getting started with the kinect for windows sdk beta from microsoft research. Beta 1 Draft version 1.1 - July 22, 2011.
- 29) Brent Mittelstadt, Howard Paul, Peter Kazanzides, Joel Zuhars, Bill Williamson, Robert Pettitt, Phillip Cain, David Kloth, Luke Rose, and Bela Musits. Development of a surgical robot for cementless total hip replacement. Robotica, 1993, 11(06):553–560.
- 30) Shimon Y. Nof. Handbook of Industrial Robotics. Wiley, 2nd edition, 1999.
- 31) PrimeSense. Primesense-3d-sensor-data-sheet. <http://www.primesense.com/en/press-room/resources/file/4-primesense-3d-sensordata-sheet>.
- 32) Guinness World Records. [http://community.guinnessworldrecords.com/ Kinect-Confirmed-As-Fastest-Selling-Consumer-electronicsDevice/blog/3376939/7691.html](http://community.guinnessworldrecords.com/Kinect-Confirmed-As-Fastest-Selling-Consumer-electronicsDevice/blog/3376939/7691.html).
- 33) Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.
- 34) J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on nov. 2011, pages 1154 –1160.
- 35) Spong, Hutchinson, and Vidyasagar. Robot modeling and control. Wiley, 1st edition, 2006.
- 36) Todor Stoyanov, Athanasia Louloudi, Henrik Andreasson, and Achin J. Lilienthal. Comparative evaluation of range sensor accuracy in indoor environments. In 5th European Conference on Mobile Robots, ECMR 2011, Orebro, Sweden, September 7-9 2011.
- 37) Jocelyne Troccaz. Computer and robot-assisted medical intervention. In Shimon Y. Nof, editor, Springer Handbook of Automation. Springer Berlin Heidelberg, 2009, pages 1451–

1466.

38) Arne Tyssø. Automatiseringsteknikk. Modelling, Analyse og syntese av reguleringsystemer. NKI forlaget, 2nd edition, 1987.

39) Mark J. P. Wolf. The Video Game Explosion, A history from pong to playstation and beyond. Greenwood Press, London, 2008.

40) ГОСТ 12.1.005-88. Міждержавний стандарт. Система стандартів безпеки праці. Загальні санітарно-гігієнічні вимоги до повітря робочої зони

41) ГОСТ 12.0.003-74 Небезпечні і шкідливі виробничі фактори. Класифікація

42) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту

43) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

44) ГОСТ 12.1.004-91. Пожежна безпека. Загальні вимоги .

45) ДБН В.2.5-67. Опалення вентиляція та кондиціонування.

46) ГОСТ 12.1.006-84. Електромагнітні поля радіочастот. Допустимі рівні на робочих місцях і вимоги до проведення контролю

47) ДБН В.2.5-28-2006. Природне і штучне освітлення.

48) ГОСТ 12.4.009-83. Пожежна техніка для захисту об'єктів. Основні види. Розміщення і обслуговування.

49) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою.

50) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів

51) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.

ДОДАТОК А.

Kinect-robot інтерфейс

```

#include <pcl/io/opensnigrabber.h>
#include <pcl/pointtypes.h>

#include <math.h>
#include <iostream>
using namespace std ;
typedef fpcl::PointXYZ PointT;
void findNextPosition ( pcl::PointCloud< PointT >::
ConstPtr cloud ) ;
void kinectGetCenterPointDistance (
pcl::PointCloud< PointT >::ConstPtr cloud ) ;
void kinectGetLastPosition ( double * pos ) ;

pcl::OpenNIGrabber * kinectInterface ; //!< A
pointer to the kinect interface
pcl::PointCloud<PointT >::ConstPtr
kinectPointCloud; //!< The last point cloud received
from the Kinect
Eigen::Vector4f kinectLastPosition ,
kinectNewPosition ; //!< Stores the center point of the
tracked plane
double kinectRotation[ 9 ] ; //!< Wanted
orientation on the end effector
boost::mutex kinectCloudMutex , kinectPointMutex;

Eigen::Matrix4f kinectTransMat ; //!< The
translation matrix to rotate from kinect coo rdinate to
robot base coordinate // ! Callback function for the
Kinect depth data
/*
* Receives a new point cloud from the Kinect and
calls kinect FindPlane
* Also prints statistics about the Kinect frame
rate
*/
void cloud_cb ( const pcl::PointCloud<PointT
>::ConstPtr & cloud ) {
if ( kinectCloudMutex.trylock ( ) ) {
kinectPointCloud = cloud;
// kinectNewCloud = true;

```

```

        findNextPosition( cloud );
        kinectCloudMutex.unlock( ) ;
    }
}
// ! Finds the next position the robot should move
to
/*
!
* Method to find the next position the robot should
move to. This method is
* up to the use r to implement . The found p o s i
t i o n should be found in the
* Kinect's coordinate system and transformed to the
robot's coordinate system
* with the function transformPoint();
*/
void findNextPosition(pcl::PointCloud< PointT
>::ConstPtrcloud ) {
    kinectGetCenterPointDistance ( cloud );
}

// !Retreive the last position from the Kinect
/*
!
* Stores the last found position from the Kinect in
the posarray.
*If the position array is busy, the last position
is returned.
* @param double pos[] - Array to store the ( x, y,
z ) coordinates in
*/
void kinectGetLastPosition( double * pos ) {
    if ( kinectPointMutex.trylock() ) {
        kinectLastPosition( 0 ) = kinectNewPosition( 0
);
        kinectLastPosition( 1 ) = kinectNewPosition( 1
);
        kinectLastPosition( 2 ) = kinectNewPosition( 2
);
        kinectPointMutex.unlock();
    }
    pos[ 0 ] = kinectLastPosition( 0 );
    pos[ 1 ] = kinectLastPosition( 1 );
    pos[ 2 ] = kinectLastPosition( 2 );
}

```



```

// ! Transforms the point from Kinect coordinate
system to robot base coordinate system
/*
!
* Stores the transformed point in
kinectPlaneCenterPoint
* The transformation matrix used is defined in
initializeKinect().
*
* @param Eigen::vector4f - the point to transform
from kinect to robot
*/
void transformPoint( Eigen::Vector4f centerPoint )
{
centerPoint = kinectTransMat * centerPoint;
if ( kinectPointMutex.trylock() ) {
kinectNewPosition = centerPoint ;
kinectPointMutex.unlock() ;
}
}
// ! Test method to only control one of the robot
axis
/*
!
* Finds the distance to the center point and
transform it to the robot coordinate
*/
void kinectGetCenterPointDistance( pcl::PointCloud<
PointT >::ConstPtr cloud ) {
Eigen::Vector4f centerPoint;
centerPoint( 0 ) = 0.0 ;
centerPoint( 1 ) = 0.0 ;
// Check for missing depth value
if ( cloud->points [ 320 * 481 ].z == cloud->points
[ 320 * 481 ].z ) {
centerPoint( 2 ) = cloud->points [ 320 * 481
].z;
transformPoint( centerPoint ) ;
}
}

//!Initializes the Kinect interface and sets all
global variables
/*!

```

```

    *Initializes a new OpenNIGrabber and all global
    variables needed for the Kinect control.
    */
    void initializeKinect() {
        printf("Starting kinect Interface\n" );
        kinectInterface = new pcl::OpenNIGrabber();
        boost::function<void          (          const
pcl::PointCloud<pcl::PointXYZ >::ConstPtr&> f1 =
        boost::bind (&cloud cb, 1 );
        boost::signals2::connection          c1          =
kinectInterface->registerCallback( f1 );
        kinectTransMat << 1.0, 0.0, 0.0, 0.00,
        0.0, 1.0, 0.0, 0.00,
        0.0, 0.0, 1.0, 0.00,
        0.0, 0.0, 0.0, 1.00;
        // Set starting position commands to default
positions
        kinectNewPosition( 0 ) = -0.5;
        kinectNewPosition( 1 ) = -0.11;
        kinectNewPosition( 2 ) = 0.45 ;
        kinectLastPosition( 0 ) = -0.5;
        kinectLastPosition( 1 ) = -0.11;
        kinectLastPosition( 2 ) = 0.45 ;
        //kinectNewCloud = false;
        kinectRotation[ 0 ] = 0; //Sets the robot tool to
pointing down
        kinectRotation[ 1 ] = 1;
        kinectRotation[ 2 ] = 0;
        kinectRotation[ 3 ] = 1;
        kinectRotation[ 4 ] = 0;
        kinectRotation[ 5 ] = 0;
        kinectRotation[ 6 ] = 0;
        kinectRotation[ 7 ] = 0;
        kinectRotation[ 8 ] = -1;

        kinectInterface->start() ;
    }
    // !
    /*!
    * Stops the Kinect process
    */
    void stopKinect() {
        printf( "Closing down Kinect\n" );
        kinectInterface->stop();
    }

```

ДОДАТОК Б.

Електронні плакати

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТУ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

МАГІСТЕРСЬКА РОБОТА НА ТЕМУ:

Дослідження комп'ютерної системи керування рухомим об'єктом

виконав:
студент групи КІ-17зм
Матюшко Катерина Ігорівна

Науковий керівник:
Барбарук Віктор Миколайович
Кандидат технічних наук, доцент

Актуальність дослідження

У зв'язку зі збільшенням кількості транспортних засобів, інтенсивності руху, а також збільшенням рівня навколишніх наших технологій стає актуальним питання автоматизації управління рухомими об'єктами (зокрема автомобілями, військовою та спеціальною технікою в екстремальних умовах).

Один із найважливіших складових елементів системи автоматичного керування є датчики комп'ютерного зору та сенсори, які можуть відновити навколишнє 3D-середовище, для прийняття наступного рішення, щодо руху.

Одна з найважливіших особливостей датчика Kinect - відновлення 3D-сцени простору, що знаходиться в полі зору пристрою. Однією з альтернативних технологій в даному напрямку є стереозрення. Воно засноване на роботі людського зору - на об'єкт спрямовані дві еталонних камери, і потім за допомогою проєктивної геометрії обчислюється відстань до об'єкта або відновлюється 3D-модель простору. Але у цієї технології є свої недоліки: неможливість коректно відновлювати простір, яке знаходиться за прозорими об'єктами, такими як вода, наприклад. Також існують лазерні далекоміри. Вони позбавлені недоліку стереозрення, але при цьому не можуть передати інформацію про яркостної характеристиці сцени, маю велике енергоспоживання і високу вартість.

Можна сказати, що сенсор Kinect є унікальним пристроєм, який можна ефективно використовувати прототипом комп'ютерного зору в робототехніці, 3D-скануванні, розпізнаванні жестів, мимики й мови, в системах спостереження і безлічі різних сфер. Основна його перевага над іншими пристроями машинного зору - ціна і доступність як для професіоналів, так і для любителів-ентузіастів.

- **Мета дослідження:**
узагальнення теорії і принципів побудови систем автоматичного управління для будь-яких рухомих об'єктів на основі комп'ютерного зору.
 - **Завдання дослідження:**

грунтуючись на теоретичній частині даної роботи, проаналізувати можливості та роль датчика Kinect в системі автоматичного керування об'єктом, що рухається.
 - **Об'єкт дослідження:**
системи отримання та опрацювання інформації о навколишньому середовищі.
 - **Предмет дослідження:**
методи отримання інформації о навколишньому середовищі за допомогою сенсорів комп'ютерного зору (зокрема – датчик комп'ютерного зору Kinect, як доступна альтернатива існуючим аналогам).
-

- **Мета дослідження:**
узагальнення теорії і принципів побудови систем автоматичного управління для будь-яких рухомих об'єктів на основі комп'ютерного зору.
 - **Завдання дослідження:**

грунтуючись на теоретичній частині даної роботи, проаналізувати можливості та роль датчика Kinect в системі автоматичного керування об'єктом, що рухається.
 - **Об'єкт дослідження:**
системи отримання та опрацювання інформації о навколишньому середовищі.
 - **Предмет дослідження:**
методи отримання інформації о навколишньому середовищі за допомогою сенсорів комп'ютерного зору (зокрема – датчик комп'ютерного зору Kinect, як доступна альтернатива існуючим аналогам).
-

ОГЛЯД ДАТЧИКУ КОМП'ЮТЕРНОГО ЗОРУ KINECT

Ігрові консолі розміщуються в різних поколіннях, щоб розділити їх, залежно від того, з якої епохи часу вони знаходяться та які технології вони використовують. У наступному розділі буде використано ці покоління підрозділів, щоб описати, як розвивалася технологія відеоігор.

Сьоме покоління є нинішнім поколінням і почалося в 2005 році за допомогою Xbox 360, незабаром слідують Playstation 3 та Nintendo Wii. Коли Xbox і PlayStation зосередили увагу на якійсь графіці HD і більшій обчислювальній потужності в консолі, ніж будь-коли раніше.

Після успіху Wii, Playstation випустила свій контролер "Move" в 2010 році. Дві основні частини контролера "Move" - це камера "Playstation Eye" і ручна паличка. Система "Переміщення" може відстежувати рух тіла гравця в трьох вимірах, використовуючи камеру та світло, випущені з палички, але на відміну від Kinect, гравець повинен тримати паличку в руці для тривимірного відстеження.

Xbox Kinect був випущений в листопаді 2010 року компанією Майкрософт для використання з консоллю Xbox 360. З його випуску і до березня 2011 року було продано понад 10 мільйонів одиниць, це було найшвидшим продажом електронного приладу взагалі. Kinect має функції контролю руху, голосових команд і розпізнавання обличчя, і є прямим конкурентом Nintendo Wii і Playstation Move.

Технологія 3D-зображень: Триангуляція

Є дві загальних методи отримання інформації про 3D-сцену, пасивне та активне поглиблення. Пасивна глибина зондування використовує просювання знімків з двох різних кутів для розрахунку глибини. Активні методи проєктують деяке світло на сцені, і використовують це для вимірювання відстані

Триангуляція може бути або активною, або пасивною [24]. У активній формі, промінь світла, як правило, лазер, проєктується на сцену в полі зору. Потім камера фіксує зображення сцени та положення світла зареєстрованого на зображенні. Відстань, D, до об'єкту точка лазеру проєктується на:

$$D = \frac{B}{f + \tan \alpha}$$

B - вихідна відстань між лазером та центром оптичної камери,

- позиція лазерної точки на зображенні,

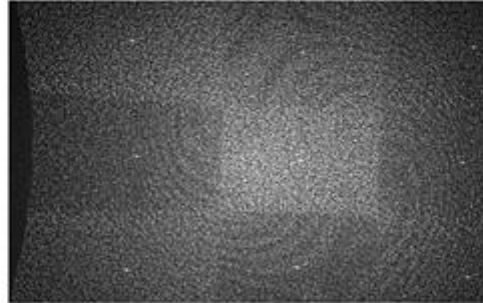
f - фокусна відстань камери

α - кут між проєктованим лазером і оптична віссю камери

Пасивна форма триангуляції полягає в тому, щоб використовувати декілька камер для прийому зображення сцени, це також відомо як стереосенс. У стереозвучі відстань обчислюється шляхом триангуляції між положеннями камери і відповідних пікселів на знімках.

Технологія 3D-зображень: Структуроване світло

Структуроване світло - це ще одна форма активної триангуляції. У структурованому світлі на сцену запроєктований відомий рисунок, розраховується глибина на основі триангуляції між відомим еталонним шаблоном і відстеженим відбитим образом.



ІЧ-шаблон Kinect

Kinect:технічні деталі

Kinect складається з інфрачервоного (ІЧ), джерела світла, датчика ІЧ-світла, RGB камери, мікрофонного масиву з чотирма мікрофонами, мотор для нахилу датчик і акселерометр для виявлення кута Kinect відносно горизонту.

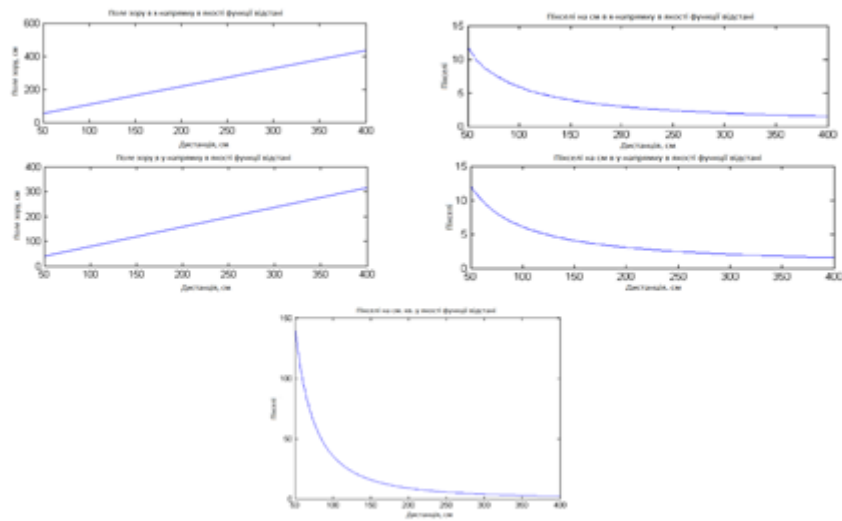
Як RGB, так і ІЧ-камери мають роздільну здатність 640x480 пікселів (VGA) і можуть доставити потік зображень 30 кадрів в секунду.

Kinect використовує лазер класу А щоб випромінювати інфрачервоне світло, і тому немає ризику пошкодження очей із застосуванням Kinect. "Мозок" в Kinect походить від системи PrimeSense PS1080

Поле зору для датчика глибини становить 57° в горизонтальному та 43° в вертикальному напрямку.

Щоб виміряти глибину у своїй області перегляду, Kinect використовує структуроване світло, або Light Coding™, як PrimeSense називає це. ІЧ-світло з відомим рисунком проєцирується на сцену, а коли світло повертається, спостережуваний рисунок використовується для розрахунку глибини.

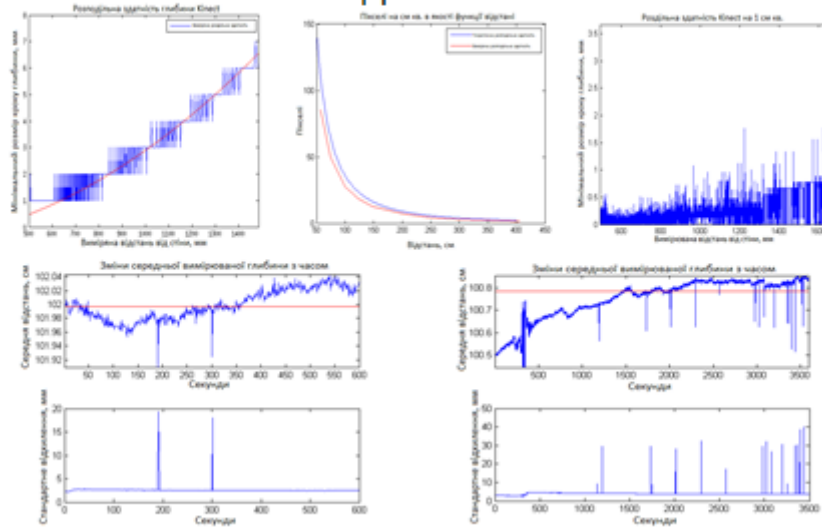
Kinect: Технічні деталі



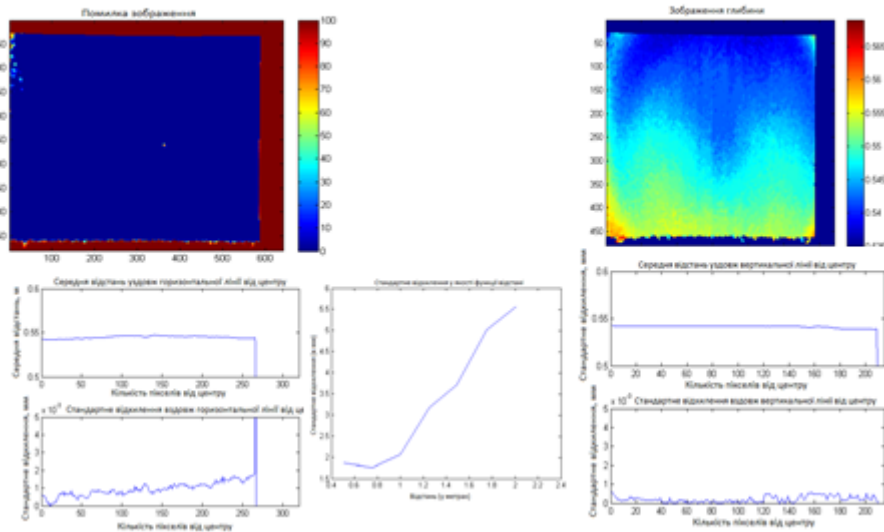
Kinect: Технічні деталі

Kinect SDK	OpenNI
Пряма підтримка управління двигуном	Немає підтримки керування двигуном
Підтримка аудіокоманд	Немає підтримки аудіо
Пропозиція повного відстеження тіла без калібрувальної позиції	Можливість відстеження тіла з калібруванням позиції
Тільки повне відстеження тіла	Підтримка ручного відстеження та вбудоване розпізнавання жестів
Тільки в Windows 7	Платформо незалежна
Може бути використаний тільки з Kinect	Може бути використаний з Kinect та Asus Xtion Pro
Лише некомерційна ліцензія	Ліцензія з відкритим кодом

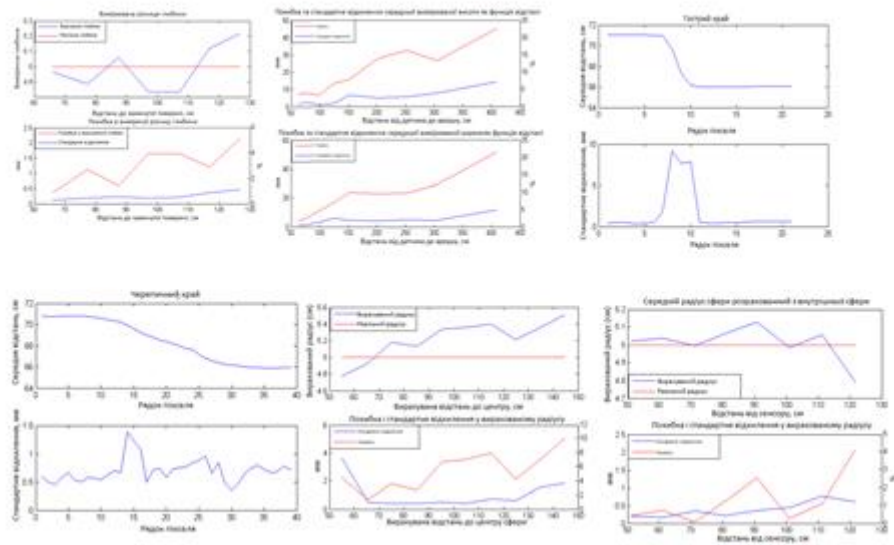
Тестування датчика Kinect: розподільна здатність і стабільність



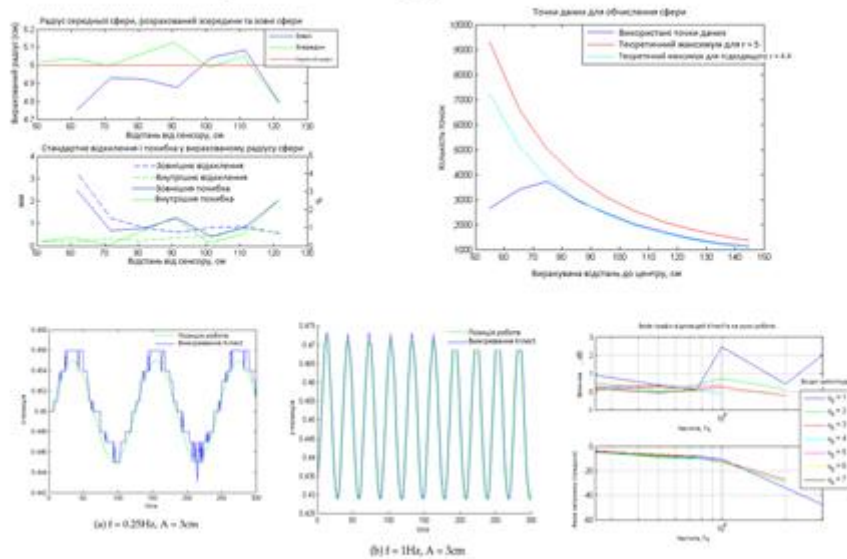
Тестування датчика Kinect: недійсні пікселі і середні значення



Тестування датчика Кінест



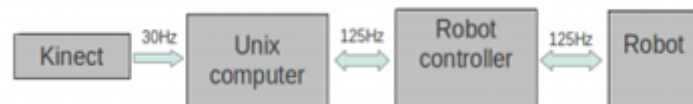
Тестування датчика Кінест



Тестування датчика Kinect: різниця між рухомих і нерухомих об'єктом



ІНТЕРФЕЙС КІНЕСТ-РОБОТ

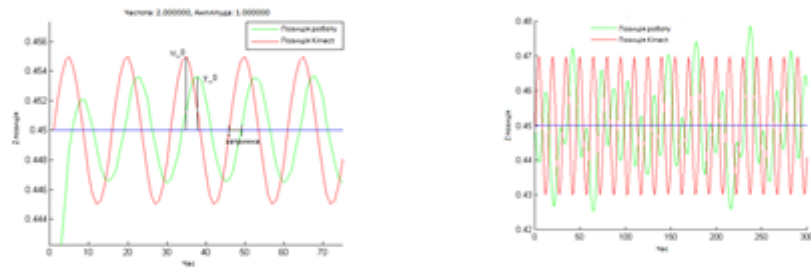


$$\text{Out} = (\text{Pd} - \text{Pc}) * \text{Gain},$$

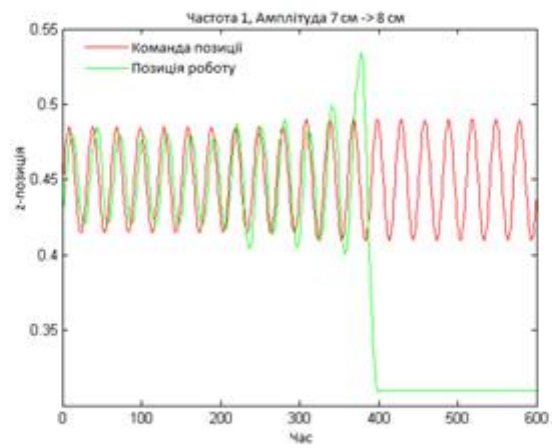
Швидкість робота контролюється пропорційним контролером. Таким чином, швидкість визначається різницею в потрібній позиції, а поточна позиція помножена на попередньо визначений коефіцієнт посилення. Pd та Pc - це бажана і поточна позиція інструмента робота.

$z(t) = A * \sin(t * 2\pi * f) / 30$, де A та f - амплітуда і частота синуса, а t - час, який вважається тактовим циклом у моделюваному процесі Kinect.

Демонстрація того, як Kinect безпосередньо контролює положення робота



Результати тесту



ВИСНОВКИ

Точність і роздільна здатність швидко зменшуються з відстанню, тому вимірювання глибини слід тримати на відстані менше 1,5 м, це дасть точність глибини > 90%.

Коли доходить до компенсації руху, Kinect не може бути використаний, це виходить з того, що Kinect не зможе побачити поверхню, і створення точної моделі на основі навколишньої поверхні буде важким. Тому сигнал іншого датчика буде кращим джерелом зворотного зв'язку для компенсації руху. Але Kinect все ще може бути використаний, щоб переконатися, що інші частини робота зберігають безпечну відстань від цілі.

Візуальні дані від Kinect найкраще використовувати для маневрування робота до правильної цілі, тоді як сигнали інших датчиків повинні використовуватися для остаточного позиціонування.
